+1/1/60+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

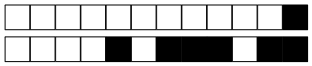
Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`



Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 7 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

☐ return 3
☐ exit 3
☐ kill 3
☐ kill -3

Question 12 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 13 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 16 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 17 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 18 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

+2/1/55+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

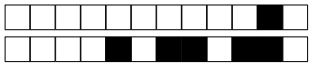
Question 1 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien



Question 3 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 4 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 7 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 8 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 9 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 10 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 11 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 12 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 13 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

+3/1/51+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

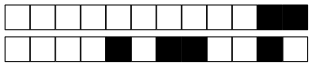
Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



+3/2/50+

Question 2 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 11 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

Question 12 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 13 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 14 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 15 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

+4/1/46+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 5 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 6 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 7 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 8 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 9 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 14 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

+5/1/42+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

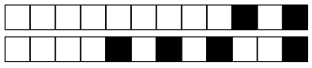
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 3 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 4 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 5 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 6 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 7 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 11 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 12 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 13 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 14 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 15 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 16 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 17 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 18 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

+6/1/37+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

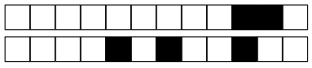
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



Question 2 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 8 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 9 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 15 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 16 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 17 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 18 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

+7/1/32+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 9 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 10 La ligne de commande `LANG=fr man ls`

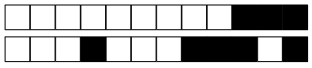
☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 11 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 12 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé



Question 13 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

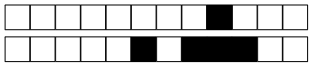
Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



+8/1/28+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 5 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

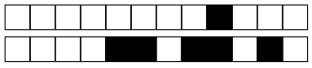
Question 6 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 7 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 12 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 13 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 14 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

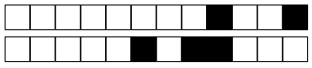
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 18 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*



+9/1/24+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

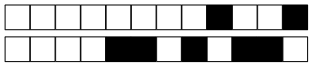
Question 6 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 7 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 10 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 11 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 12 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 13 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 14 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

+10/1/20+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 3 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 4 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

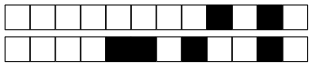
Question 5 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 6 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 7 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 8 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 9 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 10 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 11 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ commuter entre deux tâches sans attendre
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ aucune de ces propositions n'est correcte

Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

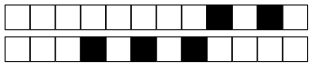
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 16 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

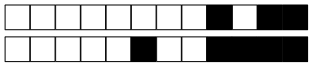
☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 18 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



+11/1/15+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

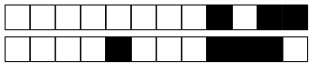
```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien



Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 5 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 10 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 11 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 15 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 16 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

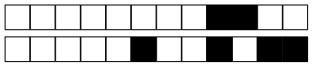
Question 17 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 18 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`



+12/1/11+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 2 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 3 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

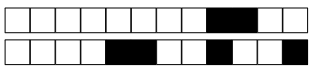
```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps

☐ garantir qu'une tâche s'exécutera dans un temps fixé

☐ commuter entre deux tâches sans attendre

☐ aucune de ces propositions n'est correcte

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

☐ return 3
☐ exit 3
☐ kill 3
☐ kill -3

Question 12 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
☐ n'affichera jamais rien
☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 17 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

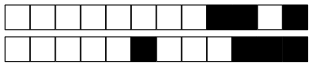
1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 18 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



+13/1/7+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 5 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 8 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 9 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 10 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 11 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 12 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 13 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

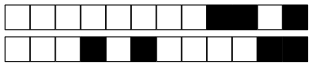
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 17 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

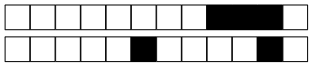
☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



Question 18 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



+14/1/2+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 3 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 6 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

+15/1/58+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

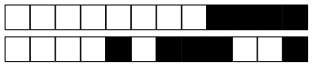
```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 5 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 6 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !	1) l'UID de l'utilisateur
b) ?	2) le GID de l'utilisateur
c) *	3) le PID du shell courant
d) $	4) le code de sortie de la dernière commande exécutée
	5) le ou les arguments courants
	6) le PID du dernier processus lancé
	7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 7 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 8 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 9 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 10 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 11 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 12 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 13 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 14 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

+16/1/54+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps

☐ garantir qu'une tâche s'exécutera dans un temps fixé

☐ commuter entre deux tâches sans attendre

☐ aucune de ces propositions n'est correcte

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu

☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide

☐ n'affichera jamais rien

☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 3 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`

☐ Lire ni `/etc/passwd` ni `/etc/shadow`

☐ Lire `/etc/passwd` et `/etc/group`

☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

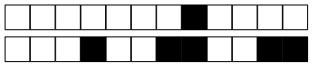
a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 14 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 15 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

+17/1/50+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 2 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 3 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 5 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 8 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 9 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

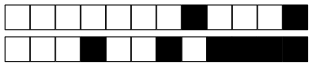
- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 12 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 13 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3



Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

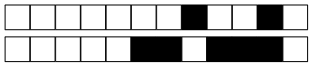
Question 17 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



+18/1/46+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

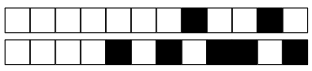
.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 8 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 9 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 16 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 17 La ligne de commande `LANG=fr man ls`

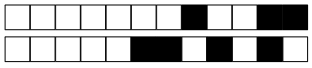
- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 18 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



+19/1/42+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

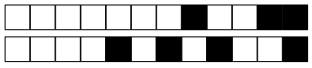
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 7 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 11 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

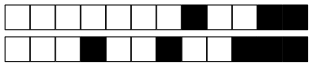
Question 12 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 13 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 14 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

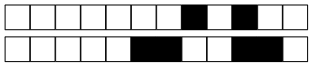
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



+20/1/38+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

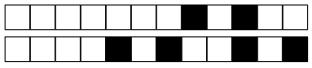
Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



Question 3 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 4 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 11 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 12 Un système est dit *temps réel* lorsqu'il peut

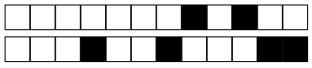
☐ exécuter deux tâches en même temps
☐ garantir qu'une tâche s'exécutera dans un temps fixé
☐ commuter entre deux tâches sans attendre
☐ aucune de ces propositions n'est correcte

Question 13 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 14 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé



Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 17 Qu'affiche le code ci-dessous ?

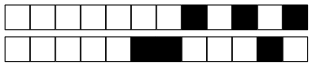
```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien



+21/1/34+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

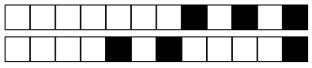
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*



Question 3 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 7 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 10 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 11 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 12 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 13 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 14 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

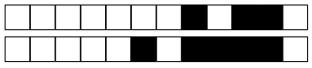
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



+22/1/30+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 2 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 10 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 11 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 12 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 13 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 14 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 15 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

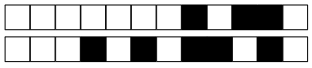
Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 17 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

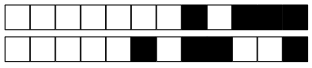
- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



+22/5/26+

Question 18 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`



+23/1/25+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

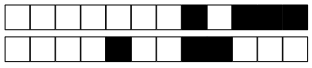
Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 2 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 3 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 4 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

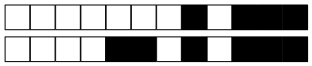
Question 5 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 6 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte



Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 9 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

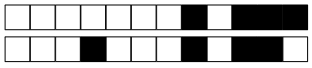
- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 10 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 11 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



Question 12 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 13 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

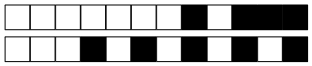
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

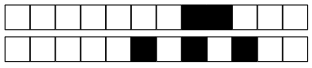
Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



+24/1/20+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0	☐0	☐0	☐0	☐0	☐0
☐1	☐1	☐1	☐1	☐1	☐1
☐2	☐2	☐2	☐2	☐2	☐2
☐3	☐3	☐3	☐3	☐3	☐3
☐4	☐4	☐4	☐4	☐4	☐4
☐5	☐5	☐5	☐5	☐5	☐5
☐6	☐6	☐6	☐6	☐6	☐6
☐7	☐7	☐7	☐7	☐7	☐7
☐8	☐8	☐8	☐8	☐8	☐8
☐9	☐9	☐9	☐9	☐9	☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 2 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 5 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 6 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 9 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 15 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

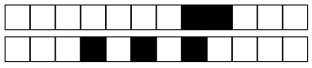
☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 16 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

☐ return 3
☐ exit 3
☐ kill 3
☐ kill -3

Question 17 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte



Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

+25/1/15+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

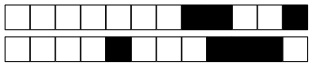
.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 5 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 7 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 8 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 11 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

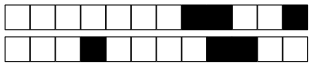
- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 16 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

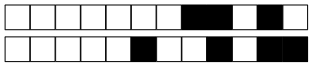
Question 17 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

Question 18 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose



+26/1/11+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

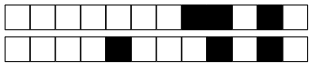
- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 2 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 3 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 4 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 5 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 6 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous
`test -d /cdrom || ls /cdrom`

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 7 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 8 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 9 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 10 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 11 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

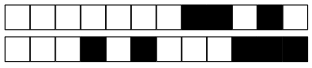
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 16 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

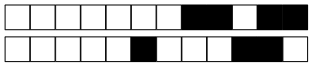
Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 18 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



+27/1/6+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 3 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 9 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 10 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 11 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 12 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 13 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 16 La variable HOME du shell représente

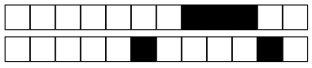
- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ commuter entre deux tâches sans attendre
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ aucune de ces propositions n'est correcte



+28/1/2+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

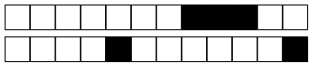
```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



Question 3 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 4 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 5 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

+28/3/60+

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 11 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

Question 12 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 13 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 14 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 15 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 16 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

+29/1/58+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps

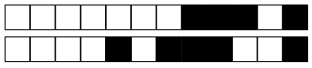
☐ garantir qu'une tâche s'exécutera dans un temps fixé

☐ commuter entre deux tâches sans attendre

☐ aucune de ces propositions n'est correcte

Question 2 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`



Question 3 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 4 Qu'affiche la ligne `echo /etc/[^e]??*`

☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
☐ Littéralement, `/etc/[^e]??*`
☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
☐ *Tout à fait autre chose*

Question 5 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

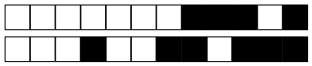
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

☐ return 3
☐ exit 3
☐ kill 3
☐ kill -3

Question 12 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose



Question 13 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

+30/1/53+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps

☐ commuter entre deux tâches sans attendre

☐ garantir qu'une tâche s'exécutera dans un temps fixé

☐ aucune de ces propositions n'est correcte

Question 2 Qu'affiche le code suivant ?

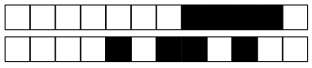
```
a=3+2; echo $(a)
```

☐ 3+2

☐ 5

☐ rien

☐ une erreur, fort probablement



Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 5 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 6 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 7 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 8 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 9 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 10 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 11 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 12 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 13 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 17 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 18 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- [ ] return 3
- [ ] exit 3
- [ ] kill 3
- [ ] kill -3

+31/1/48+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 5 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 6 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 7 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 8 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

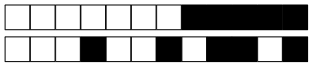
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 12 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau

☐ n'embarque que le minimum nécessaire pour démarrer

☐ n'a rien à voir avec la manière dont son noyau est formé



Question 13 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 14 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 15 La ligne de commande `LANG=fr man ls`

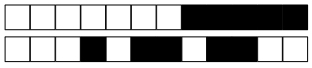
- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3



Question 18 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

+32/1/43+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

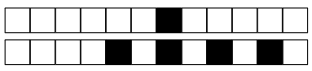
- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 3 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 6 Qu'affiche la ligne `echo /etc/[^e]??*`

☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
☐ Littéralement, `/etc/[^e]??*`
☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
☐ *Tout à fait autre chose*

Question 7 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

Question 8 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 9 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

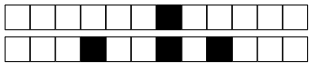
Question 12 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 13 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3



Question 14 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 15 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 17 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous
`test -d /cdrom || ls /cdrom`

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

+33/1/39+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

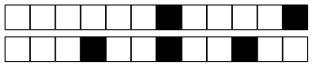
Question 11 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 12 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte



Question 13 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 14 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 15 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 16 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

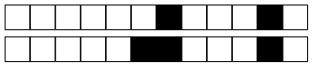
- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 17 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 18 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*



+34/1/34+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

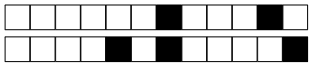
Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



Question 3 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 4 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 8 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 11 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 12 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 13 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

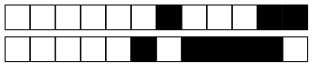
Question 17 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
☐ n'affichera jamais rien
☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 18 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé



+35/1/30+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 3 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 4 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 7 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 8 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 9 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 10 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 11 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !	1) l'UID de l'utilisateur
b) ?	2) le GID de l'utilisateur
c) *	3) le PID du shell courant
d) $	4) le code de sortie de la dernière commande exécutée
	5) le ou les arguments courants
	6) le PID du dernier processus lancé
	7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

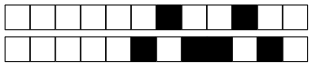
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



+36/1/26+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 8 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps
☐ garantir qu'une tâche s'exécutera dans un temps fixé
☐ commuter entre deux tâches sans attendre
☐ aucune de ces propositions n'est correcte

Question 9 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 10 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 11 La variable HOME du shell représente

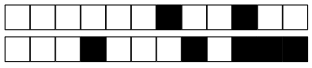
- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 12 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 13 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



Question 14 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 15 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 17 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

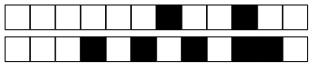
a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

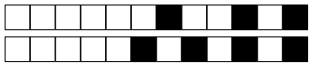
☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien





+37/1/21+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

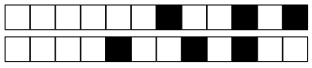
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 5 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 6 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 7 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 8 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 9 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

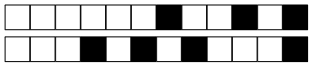
Question 16 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

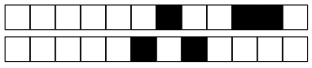
Question 17 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps
☐ garantir qu'une tâche s'exécutera dans un temps fixé
☐ commuter entre deux tâches sans attendre
☐ aucune de ces propositions n'est correcte



Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



+38/1/16+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps
☐ garantir qu'une tâche s'exécutera dans un temps fixé
☐ commuter entre deux tâches sans attendre
☐ aucune de ces propositions n'est correcte

Question 2 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 5 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 6 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 9 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 10 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 11 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

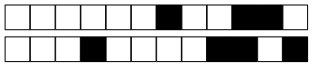
- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 16 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

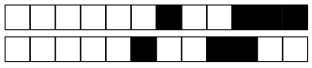
Question 17 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 18 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2
☐ 5
☐ rien
☐ une erreur, fort probablement



+39/1/12+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

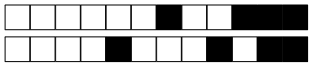
```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien



Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 5 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 8 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 9 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 12 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 13 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ commuter entre deux tâches sans attendre
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ aucune de ces propositions n'est correcte

Question 14 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

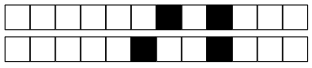
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



+40/1/8+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

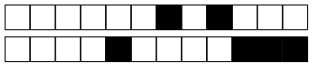
Question 1 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 2 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 5 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 6 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 7 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 11 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

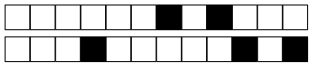
☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 12 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

Question 13 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps
☐ garantir qu'une tâche s'exécutera dans un temps fixé
☐ commuter entre deux tâches sans attendre
☐ aucune de ces propositions n'est correcte



Question 14 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 15 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

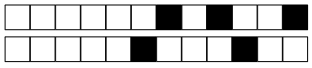
Question 17 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien



+41/1/4+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

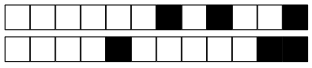
Question 1 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 2 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 12 Qu'affiche la ligne `echo /etc/[^e]??*`

☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
☐ Littéralement, `/etc/[^e]??*`
☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
☐ *Tout à fait autre chose*

Question 13 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2
☐ 5
☐ rien
☐ une erreur, fort probablement

Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 17 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 18 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

+42/1/60+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 2 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 3 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 4 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 5 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 6 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 7 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 8 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 9 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 10 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous
`test -d /cdrom || ls /cdrom`

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 11 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

+43/1/56+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 2 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 8 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 9 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 10 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 11 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 14 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 15 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 16 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 17 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 18 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

+44/1/51+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

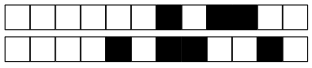
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*



Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 6 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 7 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 8 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 12 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 13 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 16 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

+45/1/47+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

☐ exécuter deux tâches en même temps

☐ commuter entre deux tâches sans attendre

☐ garantir qu'une tâche s'exécutera dans un temps fixé

☐ aucune de ces propositions n'est correcte

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2

☐ 5

☐ rien

☐ une erreur, fort probablement

Question 3 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 4 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 5 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 10 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 11 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 12 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 14 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 18 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

+46/1/43+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

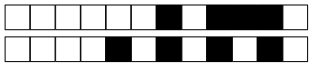
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 3 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 4 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 5 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 8 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

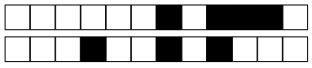
Question 10 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 11 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2
☐ 5
☐ rien
☐ une erreur, fort probablement



Question 12 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 16 La variable HOME du shell représente

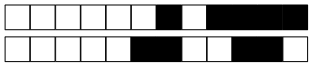
☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

☐ return 3
☐ exit 3
☐ kill 3
☐ kill -3

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 3 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 4 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 5 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 9 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 12 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 13 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

+48/1/34+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2
☐ 5
☐ rien
☐ une erreur, fort probablement

Question 2 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 3 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 6 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

Question 7 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
☐ n'affichera jamais rien
☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 8 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 14 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 15 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

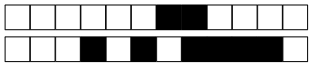
Question 16 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 17 Qu'affiche le code ci-dessous ?

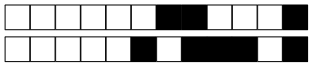
```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*



Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



+49/1/29+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 9 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 10 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 11 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 12 Qu'affiche le code suivant ?

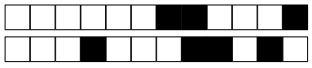
```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 13 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



Question 14 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 15 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

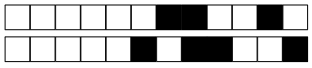
- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



+50/1/25+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 2 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Un système à noyau monolithique

☐ embarque tous ses composants dans son noyau
☐ n'embarque que le minimum nécessaire pour démarrer
☐ n'a rien à voir avec la manière dont son noyau est formé

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 12 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 13 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 14 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

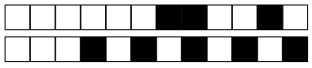
Question 16 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

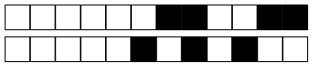
Question 17 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`



Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



+51/1/20+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 3 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 4 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 5 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 6 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 7 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 13 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

☐ Lire `/etc/passwd` et `/etc/shadow`
☐ Lire ni `/etc/passwd` ni `/etc/shadow`
☐ Lire `/etc/passwd` et `/etc/group`
☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

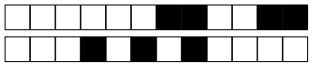
☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

Question 16 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

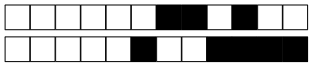
☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 18 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



+52/1/15+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 2 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

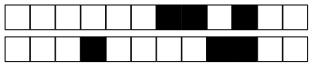
- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 10 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 11 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



Question 12 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 13 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

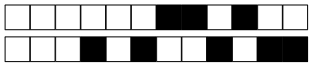
Question 15 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

+53/1/10+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 3 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 12 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 13 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 14 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 15 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 16 La ligne de commande `LANG=fr man ls`

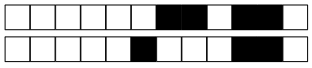
- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



+54/1/6+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

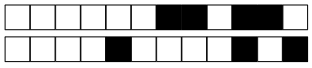
Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`



Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 5 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 6 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 7 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 16 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 17 La variable HOME du shell représente

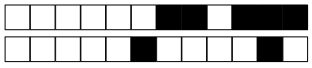
- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 18 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



+55/1/2+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 2 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 3 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 6 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 7 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

Question 8 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 9 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 10 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

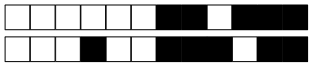
- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 11 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 12 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`



Question 13 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 14 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 15 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 17 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 18 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

+56/1/57+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

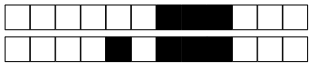
- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 3 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !	1) l'UID de l'utilisateur
b) ?	2) le GID de l'utilisateur
c) *	3) le PID du shell courant
d) $	4) le code de sortie de la dernière commande exécutée
	5) le ou les arguments courants
	6) le PID du dernier processus lancé
	7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 7 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 8 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 11 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

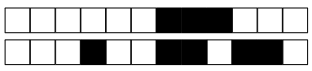
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 14 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 15 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 17 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 8 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 9 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous
`test -d /cdrom || ls /cdrom`

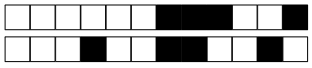
- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 10 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3



Question 12 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 13 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 16 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 17 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 18 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

+58/1/48+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 2 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 5 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 6 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 7 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 8 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 9 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 12 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 13 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 14 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

+59/1/44+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
☐ `([0-9]{3}\.){3}[0-9]{3}`
☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 3 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 9 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 10 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 11 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 12 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 13 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 14 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 15 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 17 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

+60/1/39+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 2 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 3 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 4 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 5 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 6 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 7 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

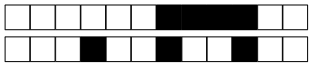
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 9 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



Question 11 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 12 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 13 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 14 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous
`test -d /cdrom || ls /cdrom`

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 15 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 17 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

+61/1/34+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

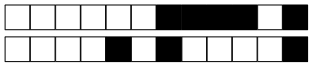
Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 2 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



Question 3 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 4 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 5 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 7 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 8 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 9 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 10 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 11 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 12 Qu'affiche le code suivant ?
`a=3+2; echo $(a)`

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 13 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 14 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

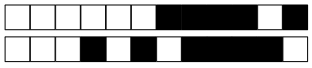
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

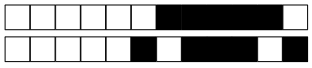
Question 17 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six



Question 18 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



+62/1/29+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 3 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 4 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 5 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!
b)	?
c)	*
d)	$

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ commuter entre deux tâches sans attendre
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ aucune de ces propositions n'est correcte

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

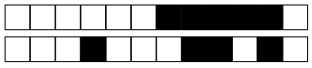
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



Question 14 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 15 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 16 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

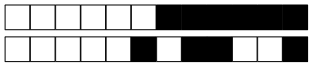
Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé



+63/1/25+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 3 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 4 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 5 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 6 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 7 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 8 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 9 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 10 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 13 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 14 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 15 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 16 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

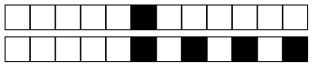
- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



+64/1/21+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 3 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 4 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 5 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 6 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 7 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 8 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 9 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 10 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

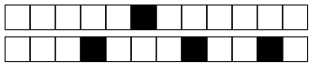
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 11 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 12 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



Question 13 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 14 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 15 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 16 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

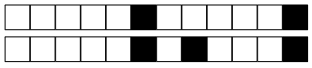
Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



+65/1/17+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 2 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 3 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1) l'UID de l'utilisateur
b)	?	2) le GID de l'utilisateur
c)	*	3) le PID du shell courant
d)	$	4) le code de sortie de la dernière commande exécutée
		5) le ou les arguments courants
		6) le PID du dernier processus lancé
		7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 9 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

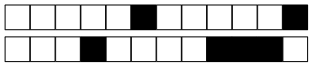
Question 10 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 11 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement



Question 12 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 13 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 14 La ligne de commande `LANG=fr man ls`

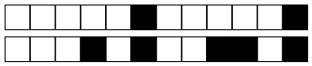
- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 15 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 16 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose



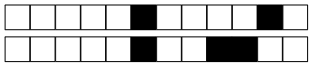
Question 17 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 18 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



+66/1/12+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 2 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 5 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

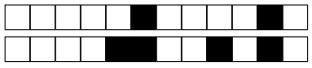
☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9



Question 9 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 10 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 11 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 12 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 13 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 14 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 15 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 16 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a)	!	1)	l'UID de l'utilisateur
b)	?	2)	le GID de l'utilisateur
c)	*	3)	le PID du shell courant
d)	$	4)	le code de sortie de la dernière commande exécutée
		5)	le ou les arguments courants
		6)	le PID du dernier processus lancé
		7)	le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

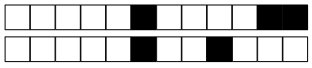
Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 18 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte



+67/1/8+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 2 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 3 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 4 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 8 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous
`a=/etc; echo "$a/"???*`

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose

Question 9 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 10 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 11 Qu'affiche le code ci-dessous ?

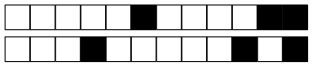
```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !	1) l'UID de l'utilisateur
b) ?	2) le GID de l'utilisateur
c) *	3) le PID du shell courant
d) $	4) le code de sortie de la dernière commande exécutée
	5) le ou les arguments courants
	6) le PID du dernier processus lancé
	7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7



Question 13 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 14 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 15 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 16 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 17 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 18 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

+68/1/4+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

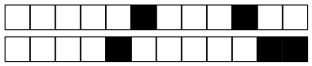
Question 1 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 2 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose



Question 3 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 4 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 5 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 6 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 7 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 8 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 9 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 10 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 11 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 12 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 13 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 14 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 15 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 16 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 17 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

☐ machin
☐ machin truc
☐ truc
☐ truc defaut
☐ *rien de tout cela*

Question 18 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

+69/1/59+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 2 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 3 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 4 La ligne de commande `LANG=fr man ls`

- ☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
- ☐ créé une variable d'environnement dont seule `man ls` aura connaissance
- ☐ n'est syntaxiquement pas correcte

Question 5 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 7 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 8 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 9 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 10 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

☐ 3+2
☐ 5
☐ rien
☐ une erreur, fort probablement

Question 11 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
☐ affiche `$a/etc`
☐ affiche `/etc/???*`
☐ affiche autre chose

Question 12 La variable HOME du shell représente

☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
☐ le répertoire par défaut de tous les utilisateur
☐ le répertoire par défaut de l'administrateur
☐ le répertoire par défaut de l'utilisateur courant
☐ autre chose

Question 13 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 14 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 15 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 16 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 17 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ commuter entre deux tâches sans attendre
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ aucune de ces propositions n'est correcte

Question 18 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

+70/1/54+

Unité 3R-IN3 (systèmes d'exploitation) **Examen du 08/06/2023**

Partie sans documents, durée 1h
Matériel électronique interdit

☐0 ☐0 ☐0 ☐0 ☐0 ☐0
☐1 ☐1 ☐1 ☐1 ☐1 ☐1
☐2 ☐2 ☐2 ☐2 ☐2 ☐2
☐3 ☐3 ☐3 ☐3 ☐3 ☐3
☐4 ☐4 ☐4 ☐4 ☐4 ☐4
☐5 ☐5 ☐5 ☐5 ☐5 ☐5
☐6 ☐6 ☐6 ☐6 ☐6 ☐6
☐7 ☐7 ☐7 ☐7 ☐7 ☐7
☐8 ☐8 ☐8 ☐8 ☐8 ☐8
☐9 ☐9 ☐9 ☐9 ☐9 ☐9

⟵ codez votre numéro d'étudiant ci-contre sans la lettre terminale, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Question 1 En supposant qu'un répertoire `/cdrom` existe sans être forcément lisible, la commande ci-dessous

```
test -d /cdrom || ls /cdrom
```

- ☐ affichera ce que contient `/cdrom` s'il est lisible ou une erreur s'il ne peut être lu
- ☐ affichera ce que contient `/cdrom` à condition qu'il ne soit pas vide
- ☐ n'affichera jamais rien
- ☐ affichera ce que contient `/cdrom` à condition qu'il puisse être lu, sinon rien

Question 2 Sur un système Linux correctement configuré, un utilisateur non privilégié peut

- ☐ Lire `/etc/passwd` et `/etc/shadow`
- ☐ Lire ni `/etc/passwd` ni `/etc/shadow`
- ☐ Lire `/etc/passwd` et `/etc/group`
- ☐ Lire `/etc/passwd` mais pas `/etc/gshadow`

Question 3 L'utilisateur `joe`, de groupe primaire `joe`, lance une commande `ls -l` dans un répertoire et obtient la sortie suivante:

```
----r--r--. 1 joe  joe       322 22 mai   16:59 un
-rw-r-xr-x. 1 joe  joe     11932 22 mai   16:59 deux
----r--rwx. 1 joe  wheel      88  2 juin  14:42 trois
-rwsr----x. 1 root root     1482 22 mai   16:59 quatre
-r-srw-r--. 1 joe  wheel  101428  2 juin  14:42 cinq
drwxr-xr-xt 5 joe  joe        56 22 mai   13:41 six
```

Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il lire ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 4 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il modifier ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 5 Sans faire quoi que ce soit de particulier, quels fichiers `joe` peut-il exécuter ?

☐ un ☐ deux ☐ trois ☐ quatre ☐ cinq ☐ six

Question 6 Coder les droits octal du fichier `cinq` dans la matrice ci-dessous:.

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

Question 7 La ligne de commande `LANG=fr man ls`

☐ créé une variable ordinaire LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement LANG permanente du shell, puis lance `man ls`
☐ créé une variable d'environnement dont seule `man ls` aura connaissance
☐ n'est syntaxiquement pas correcte

Question 8 ♣ Relier chacune des variables spéciales du shell à ce qu'elle représente

a) !
b) ?
c) *
d) $

1) l'UID de l'utilisateur
2) le GID de l'utilisateur
3) le PID du shell courant
4) le code de sortie de la dernière commande exécutée
5) le ou les arguments courants
6) le PID du dernier processus lancé
7) le répertoire courant

☐ a-1	☐ a-5	☐ b-2	☐ b-6	☐ c-3	☐ c-7	☐ d-4
☐ a-2	☐ a-6	☐ b-3	☐ b-7	☐ c-4	☐ d-1	☐ d-5
☐ a-3	☐ a-7	☐ b-4	☐ c-1	☐ c-5	☐ d-2	☐ d-6
☐ a-4	☐ b-1	☐ b-5	☐ c-2	☐ c-6	☐ d-3	☐ d-7

Question 9 Quelle est l'expression régulière étendue la plus juste pour valider une adresse IPv4 ?

- ☐ `[0-9]{,3}.[0-9]{,3}.[0-9]{,3}.[0-9]{,3}`
- ☐ `([0-9]{3}\.){3}[0-9]{3}`
- ☐ `([0-9]{1,3}\.){3}[0-9]{1,3}`
- ☐ `([0-9]{,3}\.){3}[0-9]{,3}`

Question 10 Qu'affiche le code suivant ?

```
a=3+2; echo $(a)
```

- ☐ 3+2
- ☐ 5
- ☐ rien
- ☐ une erreur, fort probablement

Question 11 Qu'affiche la ligne `echo /etc/[^e]??*`

- ☐ Les fichiers de `/etc` dont le nom débute par 2 caractères 'e'
- ☐ Littéralement, `/etc/[^e]??*`
- ☐ Les fichiers de `/etc` qui ne commencent pas par un 'e'
- ☐ Les fichiers de `/etc` d'au moins 3 caractères, et qui ne commencent pas par un 'e'
- ☐ *Tout à fait autre chose*

Question 12 Un script shell doit terminer avec le code de sortie 3. Quelle commande doit-il appeler en dernier?

- ☐ return 3
- ☐ exit 3
- ☐ kill 3
- ☐ kill -3

Question 13 Un système est dit *temps réel* lorsqu'il peut

- ☐ exécuter deux tâches en même temps
- ☐ garantir qu'une tâche s'exécutera dans un temps fixé
- ☐ commuter entre deux tâches sans attendre
- ☐ aucune de ces propositions n'est correcte

Question 14 Un système à noyau monolithique

- ☐ embarque tous ses composants dans son noyau
- ☐ n'embarque que le minimum nécessaire pour démarrer
- ☐ n'a rien à voir avec la manière dont son noyau est formé

Question 15 Qu'affiche le code ci-dessous ?

```
a=Linux-3.16
case $a in
[a-z]in*) echo machin ;;
?i*|tux) echo truc ;;
*) echo defaut ;;
esac
```

- ☐ machin
- ☐ machin truc
- ☐ truc
- ☐ truc defaut
- ☐ *rien de tout cela*

Question 16 La variable HOME du shell représente

- ☐ un chemin alternatif au répertoire `/home` si ce dernier n'existe pas
- ☐ le répertoire par défaut de tous les utilisateur
- ☐ le répertoire par défaut de l'administrateur
- ☐ le répertoire par défaut de l'utilisateur courant
- ☐ autre chose

Question 17 On veut lancer une commande `fsck -y /dev/sdc1` dont on voudrait que tout ce qu'elle produit sur sa sortie standard et sur sa sortie d'erreur standard, finisse dans un fichier `/tmp/truc`. Quelle est la bonne ligne à lancer?

- ☐ `fsck -y /dev/sdc1 < /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 > /tmp/truc 2> /tmp/truc`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 2>&1`
- ☐ `fsck -y /dev/sdc1 2> /tmp/truc 1>&2`

Question 18 En supposant que `/etc` contienne au moins 1 fichier dont le nom est long d'exactement 3 caractères, le code ci-dessous

```
a=/etc; echo "$a/"???*
```

- ☐ affiche les fichiers d'au moins 3 caractères directement rattachés à la racine
- ☐ affiche les fichiers d'au moins 3 caractères rattachés à `/etc`
- ☐ affiche `$a/etc`
- ☐ affiche `/etc/???*`
- ☐ affiche autre chose