

Correction du TD n°2

Scripts shell

Exercice 1

```
#!/bin/sh
#

for f in *; do
    case "$f" in
        *.JPEG | *.JPG) mv "$f" "${f%%.*} ".jpg
        ;;

        *.TIF) mv "$f" "${f%%.*} ".tif
        ;;
    esac
done
```

Exercice 2

Version avec shift :

```
#!/bin/sh
#

while [ "$1" ]; do
    for f in "$1"/*; do
        if [ -d "$f" ] && [ `ls "$f" | wc -l` -eq 0 ]; then
            echo "Voulez-vous supprimer $f ?"
            read rep
            if [ "$rep" = "O" ]; then
                rmdir "$f"
            fi
        fi
    done
    shift
done
```

Version non shiftée (retenir que for sans in boucle sur les paramètres) ;

```
#!/bin/sh
#

for param; do
  for f in "$param"/*; do
    if [ -d "$f" ] && [ `ls "$f" | wc -l` -eq 0 ]; then
      echo "Voulez-vous supprimer $f ?"
      read rep
      if [ $rep = "O" ]; then
        echo rmdir "$f"
      fi
    fi
  done
done
```

Exercice 3

1)

```
# conversion IP -> decimal
#
function ip2dec() {
  local ip=".1" s=0 p=1
  local droite gauche

  while [ "$ip" ]; do
    gauche="{ip%.*}"
    droite="{ip#{$gauche}.}"
    s=$((s+p*$droite))
    ip="$gauche"
    p=$((p*256))
  done

  echo $s
}
```

2)

```
# conversion decimal -> IP
#
function dec2ip () {
```

```

local i=1 n=$1 s=""

while [ $i -le 4 ]; do
    s=$((n % 256)).$s
    n=$((n / 256))
    i=$((i + 1))
done

echo ${s%?}
}

```

3)

```

# borne sup engendree par un masque
# Entree: $1 = masque en notation hexa 0x...
# Sortie: valeur de la borne sup en decimal sur stdout

```

```

function bsup () {
    local masque=`hex2dec $1`
    local p=1 s=0 i=1

    while [ $i -le 32 ]; do
        if [ $((masque % 2)) -eq 0 ]; then # bit significatif pour l'adresse IP
            s=$((s+p))
            p=$((2*p))
        fi
        masque=$((masque / 2))
        i=$((i+1))
    done

    echo $s
}

```

4)

```

# - generation d'adresses par masque

```

```

function etaler () {
    local masque=`hex2dec $1`
    local n=$2 p=1
    local s=0

    while [ $n -gt 0 ]; do

        while [ $((masque % 2)) -eq 1 ]; do # recherche d'un '0' dans le masque
            p=$((2 * p))
            masque=$((masque / 2))
        done
    done
}

```

```

done

if [ $((n % 2)) -eq 1 ]; then
    s=$((s+p))
fi

p=$((2 * p))
masque=$((masque / 2))
n=$((n / 2))
done

echo $s
}

```

5)

```

# ensemble des adresses IP generees par un masque
function ipbymask () {
    local sup=`bsup $1`
    local i=0

    while [ $i -le $sup ]; do
        dec2ip `etaler $1 $i`
        i=$((i+1))
    done
}

```