

TP programmation bash

Exercice 1 :

- I. Créer un script qui demande à l'utilisateur de saisir une note et qui affiche un message en fonction de cette note :
 - "très bien" si la note est entre 16 et 20 ;
 - "bien" lorsqu'elle est entre 14 et 16 ;
 - "assez bien" si la note est entre 12 et 14 ;
 - "moyen" si la note est entre 10 et 12 ;
 - "insuffisant" si la note est inférieure à 10.

```
#!/bin/bash
clear
echo "quel note avez vous ?"
read a

case $a in
    20|19|18|17)
        echo "très bien"
        ;;
    14|15|16)
        echo "bien"
        ;;
    12|13)
        echo "assez bien"
        ;;
    10|11)
        echo "moyen"
        ;;
    9|8|7|6|5|4|3|2|1|0)
        echo "insuffisant"
        ;;
    *)
        echo "Entrer une note valide"
        ;;
esac
```

II. Ecrire un script bash qui affiche le plus grand des entiers passes en paramètres

```
#!/bin/bash
clear
echo "vous avez entrer $# paramètre"
i=0
for param in $@
do
    tab[i]=$param
    i=$(expr $i + 1)
done

j=1
max=0
for elt in $@
do
    if [[ "$elt" -ge "${tab[j]}" ]]
    then
        if [ "$elt" -ge "$max" ]
        then
            max=$elt
        fi
    else
        if [[ "${tab[j]}" -ge "$max" ]]
        then
            max=${tab[j]}
        fi
    fi
    j=$(expr $j + 1)
done
if [[ "$max" -eq 0 ]]
then
    echo " Vous n'avez entrer aucun parametre pauvre CON "
    echo " Entrer quelques parametre Idiot █"
else
    echo " Le plus grand parametre est $max "
fi
```

Exercice 2 :

Reprenez l'exercice 1 et faites en sorte que le programme se répète tant que l'utilisateur n'a pas saisi une note négative ou 'q' (pour quitter).

Le script doit calculez le nombre de notes de saisies et en faire la moyenne tout à la fin.

```

@debian: ~ aurelien@debian: ~
1 #!/bin/bash
2 clear
3 som=0
4 b="z"
5 declare -i a
6 merde="x"
7 quit=0
8
9 compteur=-1
10 while [ "$merde" = "x" -a "$b" != "q" ]
11 do
12     echo "quel note avez vous ?"
13     echo
14     echo " Entrer une note négative ou q pour quitter "
15     echo
16     read a
17 if [[ "$a" -ge 0 ]]
18 then
19     som=$(expr "$a" + "$som")
20     compteur=$(expr $compteur + 1)
21 elif [[ "$a" -lt 0 ]]
22 then
23     compteur=$(expr $compteur + 1)
24     b="q"
25 fi
26 if [[ "$a" -eq "q" ]]
27 then
28     merde="t"
29 fi
30 clear
31 done
32 echo "la somme: $som"
33 echo "nombre de note saisi :$compteur"
34 resultat=$(echo "scale=2; $som/$compteur" | bc)
35 echo "le resultat : $resultat"
36

```

Exercice 3 :

Créer un script qui prend un nombre en saisie et l'élève à sa propre puissance. C'est un peu le même principe que la factorielle mais cette fois, **l'usage de la boucle for est imposé.**

```

@debian: ~ aurelien@debian: ~
1 #!/bin/bash
2 clear
3 declare -i entier
4 echo "Entrer un entier"
5 read entier
6 resultat=1
7
8 for var in $(seq $entier)
9 do
10     resultat=$(expr "$resultat" \* "$entier")
11 done
12
13 echo "$entier a la puissance $entier = $resultat"

```

Exercice 4 :

Reprenez uniquement la version I de l'exercice. La note devra être donnée en paramètre ou bien saisie en cas d'absences d'arguments. La comparaison de la note devra être faite dans une fonction *appreciation()*.

```
#!/bin/bash
clear
if [[ $# -eq 0 ]]
then
    echo "quel note avez vous ?"
    read a
fi

appreciation ()
{
    case $a in
        20|19|18|17)
            echo "très bien"
            ;;
        14|15|16)
            echo "bien"
            ;;
        12|13)
            echo "assez bien"
            ;;
        10|11)
            echo "moyen"
            ;;
        9|8|7|6|5|4|3|2|1|0)
            echo "insuffisant"
            ;;
        *)
            echo "Entrer une note valide"
            ;;
    esac
}

if [[ $# -eq 0 ]]
then
    appreciation $a
else
    a=$1
    appreciation $a
fi
```

Exercice 5 :

Créer un script qui vous propose le menu suivant :

- 1 - Vérifier l'existence d'un utilisateur
- 2 - Connaître l'UID d'un utilisateur
- q - Quitter

```
an@debian: ~
x aurelien@debian: ~
1 #!/bin/bash
2 clear
3 echo
4 echo " 1- Vérifier l'existence d'un utilisateur "
5 echo " 2- Connaître l'UID d'un utilisateur"
6 echo " q- Quittez"
7 echo
8 echo "Votre choix: "
9 read choix
10
11 saisi_utilisateur ()
12 {
13     echo " Entrer le nom de l'utilisateur "
14     read nom
15     export nom
16 }
17
18 verification_nom ()
19 {
20     resulVerNom=`cat /etc/passwd | cut -d: -f1 | grep $1`
21     if [ "$resulVerNom" = "$1" ]
22     then
23         echo " l'utilisateur $nom existe "
24     else
25         echo " l'utilisateur $nom n'existe pas "
26     fi
27 }
28
29 verification_uid ()
30 {
31     resulVerNom=`cat /etc/passwd | cut -d: -f1 | grep $1`
32     if [ "$resulVerNom" = "$1" ]
33     then
34         resultVerUid=`cat /etc/passwd | grep "$1" | cut -d: -f3`
35         echo " l'UID de $1 est $resultVerUid "
36     else
37         echo " l'utilisateur $nom n'existe pas "
38     fi
39 }
40
41 case $choix in
42     1)
43         saisi_utilisateur
44         verification_nom $nom
45         ;;
46     2)
47         saisi_utilisateur
48         verification_uid $nom
49         ;;
50     q)
51         exit
52         ;;
53     *)
54         echo " saisissez ce qu'on vous demande! Conard "
55         ;;
56 esac
```

Exercice 6 :

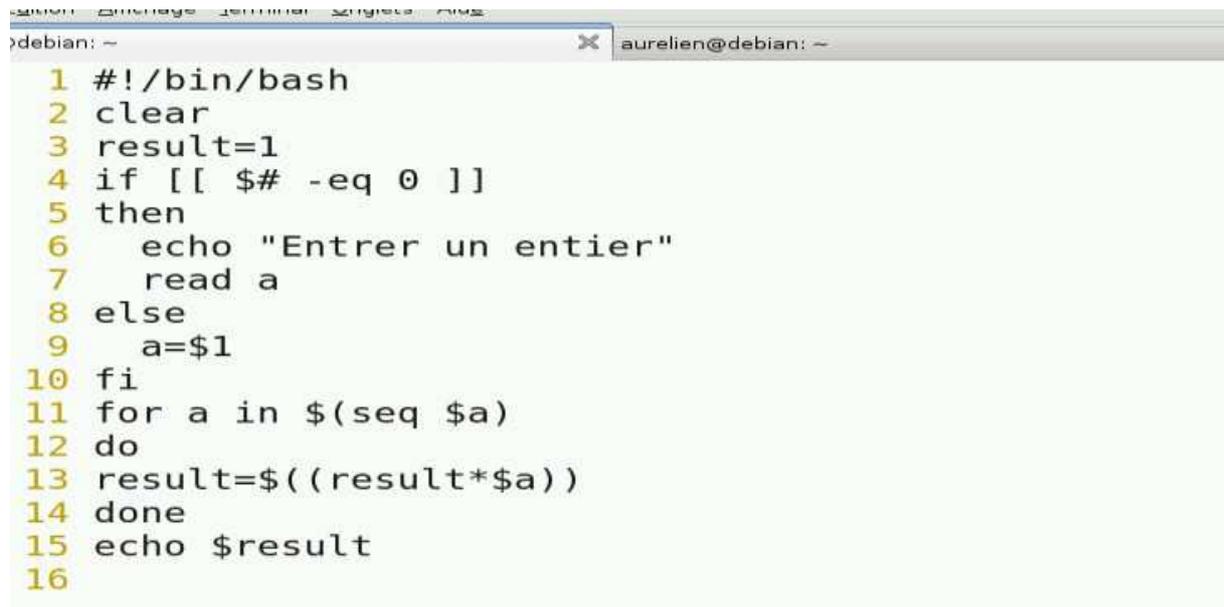
Créer un script dans lequel deux nombres opérands et un signe opérateur (+-*/) devront être donnés en paramètres, ou saisis. Le script doit réaliser l'opération souhaitée.

Le calcul devra être fait à l'aide d'une fonction **calcul ()**.

```
en@debian: ~
x aurelien@debian: ~
1 #!/bin/bash
2 clear
3 calcul ()
4 {
5     if [ "$3" = "+" ]
6     then
7         resultl=$(expr $1 + $2)
8     elif [ "$3" = "-" ]
9     then
10        resultl=$(expr $1 - $2)
11    elif [ "$3" = "/" ]
12    then
13        resultl=$(echo "scale=2; $1/$2" | bc)
14    else
15        resultl=$(( $1*$2 ))
16    fi
17    echo " resultat: $resultl"
18 }
19
20 if [[ $# -eq 3 ]]
21 then
22     calcul $1 $2 $3
23 elif [[ $# -eq 0 ]]
24 then
25     echo "Entrer le premier entier"
26     read a
27     echo "Entrer le second "
28     read b
29     echo "Entrer l'opérateur"
30     read c
31     set $a $b $c
32     calcul $1 $2 $3
33 elif [[ $# -eq 9 ]]
34 then
35     calcul $1 $2 \*
36 else
37     echo "Imbecile respecter ce script en passant les arguments correct"
38 fi
```

Exercice 7 :

Créer un script qui permet de calculer et d'afficher la factorielle d'un nombre donné en paramètre (ou saisi en cas d'absence de paramètres).



```
1 #!/bin/bash
2 clear
3 result=1
4 if [[ $# -eq 0 ]]
5 then
6     echo "Entrer un entier"
7     read a
8 else
9     a=$1
10 fi
11 for a in $(seq $a)
12 do
13 result=$((result*$a))
14 done
15 echo $result
16
```

Exercice 8 :

Créer un script qui doit calculer le nombre de fichiers standard, de sous-répertoires, et d'exécutables d'un répertoire quelconque qui sera donné en paramètre (ou saisis en cas d'absence du paramètre).

```
1 @debian: ~
2 x aurelien@debian: ~
3 x aurelien@debian
4 1 #!/bin/bash
5 2 clear
6 3
7 4 fichier_normal=0
8 5 sous_rep=0
9 6 executable=0
10 7
11 8 if [[ $# -eq 0 ]]
12 9 then
13 10     echo "Entrer le chemin du repertoire: "
14 11     read chemin
15 12 else
16 13     chemin=$1
17 14 fi
18 15
19 16 cd $chemin
20 17
21 18 for param in *
22 19 do
23 20     if [ -x $param ]
24 21     then
25 22         executable=$(expr $executable + 1)
26 23     elif [ -d $param ]
27 24     then
28 25         sous_rep=$(expr $sous_rep + 1)
29 26     elif [ -f $param ]
30 27     then
31 28         fichier_normal=$(expr $fichier_normal + 1)
32 29     fi
33 30 done
34 31 echo
35 32 echo " $fichier_normal fichier normal"
36 33 echo " $sous_rep sous-repertoire"
37 34 echo " $executable fichier executable"
38 35
```

Exercice 9 :

Créer un script qui devra enregistrer à l'aide d'un tableau, un nombre d'entiers donné en paramètre (ou en saisie) puis trier ceux-ci dans l'ordre croissant dans ce même tableau (sans passer par un autre) et enfin afficher le contenu du tableau (ordonné) sur la sortie standard.

```
lien@debian: ~/Desktop/script aurelien@debian:
1 #!/bin/bash
2 clear
3
4 fonction_tri ()
5 {
6   for tri in $(seq $a)
7   do
8     itemp=${tab[tri]}
9     for i in $(seq $a)
10    do
11      if [[ "${tab[tri]}" -lt "${tab[i]}" ]]
12      then
13        echange=${tab[i]}
14        tab[i]=${tab[tri]}
15        tab[tri]=$echange
16      fi
17    done
18  done
19  for fin in $(seq $a)
20  do
21    echo "${tab[fin]}"
22  done
23 }
24
25 if [[ $# -eq 0 ]]
26 then
27   echo " combien d'entier voulez vous entrer dans le tableau? "
28   read a
29   for var in $(seq $a)
30   do
31     echo " entrer l'element $var "
32     read tab[var]
33   done
34 else
35   a=$#
36   tab[1]=$1
37   tab[2]=$2
38   tab[3]=$3
39   tab[4]=$4
40   tab[5]=$5
41   tab[6]=$6
42   tab[7]=$7
43   tab[8]=$8
44   tab[9]=$9
45 fi
46
47 fonction_tri
48
49
```

Pour cet exercice, si on les valeurs sont données en paramètre, seul les neuf premières seront pris en compte. Si par contre on choisi de saisir, ca peut aller à l' infini. Donc à vous de faire le choix selon votre convenance

Exercice 10 :

Créer un script qui prend en paramètre ou en saisie le nom d'un fichier contenant le nom des élèves et leurs trois notes. Le script devra : afficher les noms des élèves, puis calculer et afficher la moyenne de chaque élève

Voici comment se présente le fichier :

```
Durand 12 9 14
Lucas 8 11 4
Martin 9 12 1
```

```
liien@debian: ~/Desktop/script aurelien@debian: ~/Desktop/script
1 #!/bin/bash
2
3 if [[ $# -eq 0 ]]
4 then
5     echo "Entrer le chemin complet pour acceder au fichier "
6     echo " Exemple /home/aurelien/Desktop/tp/fichier.txt "
7     echo
8     read a
9     clear
10 else
11     a=$1
12     clear
13 fi
14 echo
15 echo "Contenu du fichier $a"
16 echo
17 cat $a
18 nom1=`cat $a | cut -d" " -f1 | grep aurelien`
19 note1Nom1=`cat $a | cut -d" " -f1,2 | grep aurelien | cut -d" " -f2`
20 note2Nom1=`cat $a | cut -d" " -f1,3 | grep aurelien | cut -d" " -f2`
21 note3Nom1=`cat $a | cut -d" " -f1,4 | grep aurelien | cut -d" " -f2`
22 echo
23 echo
24 echo
25 nom2=`cat $a | cut -d" " -f1 | grep pascal`
26 note1Nom2=`cat $a | cut -d" " -f1,2 | grep pascal | cut -d" " -f2`
27 note2Nom2=`cat $a | cut -d" " -f1,3 | grep pascal | cut -d" " -f2`
28 note3Nom2=`cat $a | cut -d" " -f1,4 | grep pascal | cut -d" " -f2`
29
30 nom3=`cat $a | cut -d" " -f1 | grep anael`
31 note1Nom3=`cat $a | cut -d" " -f1,2 | grep anael | cut -d" " -f2`
32 note2Nom3=`cat $a | cut -d" " -f1,3 | grep anael | cut -d" " -f2`
33 note3Nom3=`cat $a | cut -d" " -f1,4 | grep anael | cut -d" " -f2`
34
35 nom4=`cat $a | cut -d" " -f1 | grep collens`
36 note1Nom4=`cat $a | cut -d" " -f1,2 | grep collens | cut -d" " -f2`
37 note2Nom4=`cat $a | cut -d" " -f1,3 | grep collens | cut -d" " -f2`
38 note3Nom4=`cat $a | cut -d" " -f1,4 | grep collens | cut -d" " -f2`
39
40 somme1=$(expr $note1Nom1 + $note2Nom1 + $note3Nom1)
41 somme2=$(expr $note1Nom2 + $note2Nom2 + $note3Nom2)
42 somme3=$(expr $note1Nom3 + $note2Nom3 + $note3Nom3)
43 somme4=$(expr $note1Nom4 + $note2Nom4 + $note3Nom4)
44
45 moyenne1=$(echo "scale=2; $somme1/3" | bc)
46 moyenne2=$(echo "scale=2; $somme2/3" | bc)
47 moyenne3=$(echo "scale=2; $somme3/3" | bc)
48 moyenne4=$(echo "scale=2; $somme4/3" | bc)
49
50 echo "$nom1 a eu une moyenne de: $moyenne1"
51 echo
52 echo "$nom2 a eu une moyenne de: $moyenne2"
53 echo
54 echo "$nom3 a eu une moyenne de: $moyenne3"
55 echo
56 echo "$nom4 a eu une moyenne de: $moyenne4"
```

Posez vos question et suggestions à :

aurelien.lemen@gmail.com

aurelien-lemen.overblog.com

<http://www.lopensource.org>