



Classe:.....

Nom:.....

Groupe: .....

Prénom:.....

**CONTRÔLE : SYSTÈME D'EXPLOITATION 4  
3IIR**

*Rappels:*

- la syntaxe de "case" : case "\$variable" in  $\epsilon$  value) traitements  $\epsilon$  ;;  $\epsilon$  esac

A- Peut-on accéder au 10<sup>ème</sup> argument d'un script à l'aide des variables positionnelles? Si oui, comment? Sinon pourquoi? (2pt)

OUI / NON :JUSTIFICATION :

OUI :  `${10}`.....

B- Quel est le script qui s'exécute automatiquement à chaque fois que je lance une nouvelle instance du shell bash ? (1pt)

`~/.bash_profile      ~/.bash_login      ~/.bash      ~/bash.sh  
 /etc/profile       ~/.bashrc      ~/.bash_init      /etc/init.d`

C- Soit le fichier **f** contenant le texte suivant:

**3IIR1,Programmation Java,G2,j3,10:45,12:45**

Quel sera le résultat de la commande suivante: (2pt)

`$ cat f | cut -d "," -f5-6 | tr 05 10`

`11:40,12:40`.....

D- Nous avons un fichier exécutable "`~/bin/controle`" qui affiche la longueur du premier argument qu'on lui passe.

D-1 Donnez le code source de ce fichier : (2pt)

`echo ${#1}`.....

D-2 Donnez l'expression qui permettra d'ajouter le dossier `~/bin/` à la variable PATH : (2pt)

`export PATH=$PATH:~/bin/`.....

D-3 Quel sera l'effet de cette opération (ajout du dossier dans PATH) sur le fichier `~/bin/controle` ? (1pt)

*On peut l'exécuter comme une commande depuis n'importe quel dossier sans spécifier son chemin*

E- Nous lançons parfois le bash de la manière suivante:

`$ bash -o posix`

E-1 à quoi cela peut servir? (1pt)

*à restreindre bash à la norme posix, pour pouvoir tester la portabilité des scripts.....*

E-2 donnez une autre syntaxe pour faire la même chose sans ouvrir un nouveau bash. (1pt)

`set -o posix.....`

F- La commande `getopts` dans `bash` permet de lire les options du script.

F-1 Donnez la boucle nécessaire pour consommer les options suivantes et les mettre dans des variables du même nom : (4pt)

-a sans valeur, -b sans valeur, et -c avec valeur

```
while getopts ":abc:" o; do
  case "$o" in
    a)  a=true
        ;;
    b)  b=true
        ;;
    c)  c=${OPTARG}
        ;;
    *)  echo "option non supportée" >&2 && exit 1
        ;;
  esac
done
```

.....  
.....  
.....  
.....

F-2 La commande `getopts` génère des variables dans le shell, lesquelles parmi les suivantes : (2pt)

GETOPTS    OPTARGS    **OPTARG**    OPTVAL    HOME    **OPTIND**    OPTSATE  
SHELL    MAILTO    OPTIN    VALUES    PATH    OPTTAG    OPTAGE

F-3 Entourez la/les autres commandes qui permettent aussi de traiter les options du script : (2pt)

`readopts`    `read`    `get`    **`getopt`**    `giveopt`    `alias`