Classe:..... Nom:.....
Groupe: ..... Prénom:.....

CONTRÔLE : SYSTÈME D'EXPLOITATION 4
3IIR

Rappels:
- l'option +%D de la commande date permet d'afficher la date au format 27/04/2022.
- "let" permet d'évaluer une expression arithmétique simple.
- Le trou noir : /dev/null

1ère partie : compréhension

A- Soit un script ~/bin/monscript.sh qui contient le code suivant :

```
#!/bin/bash
[ $# -ne 2 ] && echo "USAGE: `basename $0` fichier date" >&2 && set "test.txt" $(date +%D)
var1=$1 ;echo ${var1/.txt/}; shift 2; echo $#; echo ${var1:-"NAN"}
```

Sachant que notre script est exécutable et qu'on le lance sous bash de la manière suivante: $ ~/bin/monscript.sh

1. Quel serait le résultat affiché à la console? *(4pt)*

```
USAGE: monscript.sh fichier date
test
0
test.txt
```

2. Quel serait le résultat affiché si on remplace **set "test.txt" $(date +%D)** par **set -x** *(2pt)*

```
USAGE: monscript.sh fichier date
+ var1=
+ echo

+ shift 2
+ echo 0
0
+ echo NAN
NAN
```

3. Donnez la ligne qui permettra d'exécuter ce script en redirigeant les messages d'erreurs vers le trou noir.*(3pt)*

```
$ ~/bin/monscript.sh 2> /dev/null
```

4. Donnez la ligne crontab qui permettra d'exécuter ce script à chaque redémarrage de la machine en renvoyant la sortie standard et les erreurs vers un fichier **monfichier.log** *(3pt)*

```
@reboot ~/bin/monscript.sh &> monfichier.log
```

B- Citez 2 différences entre les expressions `./script.sh` et `. script.sh` *(2pt)*

./script.sh : c'est une execution, le script doit être exécutable et contenir la ligne shebang

. script.sh : c'est un sourcing, c'est-à-dire une interprétation dans le shell courant, il pourra donc utiliser les variables locales déjà déclarées dans le shell courant et sera interprété dans le même shell (pas besoin de ligne shebang)

2ème partie : interprétation

Quel serait les résultats intermédiaires et le résultat final à l'affichage après l'exécution du script suivant:*(6pt)*

```
#!/bin/bash
var1=`(var2=33; let var1=3*$var2; echo 1)`
var3="${var1:+00}.${var2:-$var1}"
var1="${var1}.$var3"
echo "${var1/./5}.${#var1[0]}"
```

```
var 1 contient: 1
var 3 contient: 00.1
var 1 contient: 1.00.1
```

Affichage final :

- ☐ 991500.991.1
- ☐ 991500.991.10
- ☒ **1500.1.6**
- ☐ 1500.33.7
- ☐ 1500533.7
- ☐ autre :

.....