

 <b>Module : LPI 101 102</b>	<b>Enseignante : OUESLATI H��la</b>
<b>TP 07 Scripts Shell</b>	<b>Niveau : RSI 21</b>
	<b>Ann��e universitaire : 2024/2025</b>

### Exercice 1

Etant donn   le script Shell myfirstscript suivant :

```
ISET= "SILIANA"
```

```
Echo ISET
```

```
Echo $ISET
```

```
Echo $0
```

```
Echo $1
```

```
Echo $#
```

```
Echo $@
```

```
Shift
```

```
Echo $2
```

```
Echo $#
```

```
Set abc def ghi
```

```
Echo $3
```

```
Echo $@
```

```
A=aaa
```

```
B=bbb
```

```
C=ccc
```

```
D=ddd
```

```
Echo \ $A\ "BCD
```

```
Echo ` $A`BCD`
```

```
Echo "$A- -B++$CD"
```

```
Echo "$A- -B++${C}D"
```

```
D="Aujourd\'hui, nous sommes les `date` "
```

```
Echo $D
```

Donner les affichages nécessaires après avoir exécuter le script en question comme suit :

```
$myfirstscript a1 b1 c3 d4 e5
```

### Exercice 2

- 1) Ecrire un script Shell liste qui affiche le message "voici les fichiers du répertoire courant : " puis appeler la commande **ls**.
- 2) Ecrire un script affiche qui affiche le contenu du fichier nomfichier (avec cat ou more).  
Le nom du fichier sera donné sur la ligne de commande.
- 3) Améliorer le script affiche pour afficher un message d'erreur, si le fichier n'existe pas.

### Exercice 3

Ecrire un programme qui calcule la moyenne, le produit et la somme d'une série de valeurs.

- 1) Donner en paramètres
- 2) Lu à partir du clavier

### Exercice 4

Ecrire un script shell createfile ayant pour syntaxe (\$createfile nom n) permettant de créer après confirmation le fichier nom\_n.

Exemple :

```
$createfile test 3 : Permet de créer le fichier test_3
```

### **Exercice 5**

Ecrire un script `del` (une autre version de `rm`) qui déplace les fichiers à supprimer dans un répertoire temporaire plutôt que les détruire immédiatement et permet de le vider périodiquement.

### **Exercice 6**

Ecrire un script shell qui affiche le nombre de ses paramètres , la liste de ses paramètres ainsi que son nom.

### **Exercice 7**

Ecrire un script shell qui accepte comme paramètre un nom. Il doit renseigner sur le type du fichier dont le nom est spécifié, ses autorisations d'accès pour l'exécuter et s'il est vide ou non.

### **Exercice 8**

Ecrire un script shell affichant la liste des noms de logins des utilisateurs définis dans `/etc/passwd` ayant un UID supérieur à 500.

Indication : `for in $(cat /etc/passwd)` permet de parcourir les lignes du fichier.

### **Exercice 9**

Ecrire un script shell pour tester si un fichier est ordinaire, un répertoire ou spécial.

### **Exercice 10**

Ecrire une commande `lsdir` qui affiche la liste des sous répertoires d'un répertoire donné (par défaut le répertoire courant).

### **Exercice 11**

Ecrire une nouvelle version de la commande `lsdir` qui peut admettre plusieurs arguments.

**Exercice 12**

Ecrire un script shell qui permet de saisir une paire de login et de mot de passe et les sauvegarder dans un fichier users.txt sous la forme login :password.

Pour arrêter le traitement vous taper STOP.

**Exercice 13**

Ecrire un script shell « saisie » qui demande la saisie d'un login (exemple esprit) et un mot de passe (exemple ingénieur) et qui affiche à l'écran « votre nom est 'esprit' et votre mot de passe est 'ingénieur' ».

**Exercice 14**

Ecrire un script shell « verif\_rep » qui détermine si le nom de fichier fourni est un répertoire.

**Exercice 15**

Ecrire un script « choix » qui demande la saisie d'un choix et qui boucle jusqu'à ce que ce choix soit « 1 » ou « 2 »