

Rappels sur if ... fi

- Sachant que la commande « who » permet de savoir qui est actuellement connecté sur une machine et que la commande « ypcat passwd » permet de savoir qui est enregistré (si NIS n'est pas installé, faire un simple « grep /etc/passwd » ou encore « getent passwd » si votre système est connecté à un annuaire LDAP), écrivez le programme shell « quidonc » qui admet un paramètre « marcel », et qui affiche un des trois messages suivant :

```
marcel est inconnu du système
marcel est actuellement connecté
marcel est connu mais actuellement non connecté
```

For ... do ...done

- En utilisant une structure de boucle « for », écrivez le programme « voyelle » qui affiche la liste de toutes les ressources d'un répertoire dont le nom commence et finit par la même voyelle. Le répertoire devra être passé en paramètre.

```
Ex : voyelle ~dupond
~dupond/aa ~dupond/agaga
~dupond/ici
~dupond/otto ~dupond/ouillouillo
```

- Testez ce programme sur le répertoire « ~bascou/tpunix/voyelles »

Rappel: pour éviter les messages d'erreur à l'écran : 2>/dev/null (voir cours)

For ... do ...done

- Écrivez un programme shell « boucle1 » permettant d'afficher à l'écran les divers paramètres, à raison d'un seul par ligne (boucle sur \$*):

```
Ex : boucle1 x 12 toto
x est un paramètre
12 est un paramètre
toto est un paramètre
```

- Modifiez « boucle1 » en « boucle2 » qui donne exactement le même résultat sans utiliser la variable de boucle dans le programme (utilisez à la place « shift » et \$1)
- Dans la boucle principale des deux programmes, insérez la ligne suivante :
echo "nb param: \$#, liste param: \$*"
Exécutez et concluez.

While ... do ... done

Écrivez le programme « neverending » qui affiche en boucle un message « toto existe », tant que le fichier « toto » existe dans le répertoire courant. Le nom du fichier (toto dans cet exemple) devra être passé en paramètre.

```
Ex : neverending fictest
fictest existe
fictest existe
...
```

Lancez ce programme en boucle infinie sur un fichier existant. Pendant ce temps, ouvrez une autre fenêtre shell, et détruisez le fichier testé (rm toto). Concluez ...

Read (saisie de donnée sur l'entrée standard)

- Écrivez un programme « accueil » qui demande son nom à un utilisateur, et qui le salue poliment ...

```
Ex : accueil
Comment t'appelles tu ?
→ Marcel (saisie au clavier)
Bonjour Marcel
```

- Modifiez ce programme pour qu'il ne demande le nom que s'il n'a pas été passé en paramètre :
Ex : `accueil2 Jules`
 `Bonjour Jules`
Ex : `accueil2`
 `Comment t'appelles tu ?`
 `...`
- Créez un fichier « data » contenant une seule ligne : « Edouard ». Relancez « accueil » en redirigeant l'entrée standard sur le fichier « data ». Concluez.

Case ... esac

- Écrivez un programme « menu » qui demande une valeur à l'utilisateur et qui affiche un message différent selon la valeur saisie.