

Série 4

Exercice 1

Créez un script interactif, en utilisant la structure "select", qui permet de tester les structures "for", "while" et "until". Lors du lancement du script, il devra proposer ce menu de 4 choix ("select") à l'utilisateur:

```
Quelle structure souhaitez-vous tester?  
1. for  
2. while  
3. until  
4. quitter
```

Dans la structure "select", utiliser la structure "case" pour évaluer la valeur choisie par l'utilisateur.

Pour le test des structures "for", "while" et "until", demander à l'utilisateur un nombre et itérez de 0 à ce nombre en affichant à chaque itération la valeur courante du compteur puis revenir au menu.

Exemple:

```
Test de la structure 'for'. Entrez le nbre d'iterations: 3  
for 0  
for 1  
for 2
```

Exercice 2

Créez un script nommé "sort_and_archive" dont le fonctionnement est le suivant:

Le script pourra avoir 2 paramètres différents:

1. un répertoire quelconque, obligatoire.
2. un nom de fichier archive à créer, optionnel

Faites attention à mettre en place des contrôles sur ces 2 paramètres afin d'éviter des erreurs en cas de mauvais paramètres ou de paramètres inexistantes

Le but du script sera d'analyser le contenu du répertoire donné en paramètre 1 et de trier ce contenu en se basant sur les extensions des fichiers contenus.

Avant de commencer le tri, un nouveau dossier devra être créé, au même endroit que l'original, dont le nom sera le nom du dossier d'origine, suivi de "-sorted".

Pour chacune des extensions de fichiers trouvées dans le répertoire d'origine, un dossier correspondant devra être créé à l'intérieur du dossier de destination et tous les fichiers ayant cette extension devront y être hard-linkés afin d'éviter d'utiliser de la place inutilement sur le disque dur.

Par exemple:

tous les fichiers se terminant par .gif devront être déplacés dans le répertoire "gif", tous les fichiers se terminant par .xls devront être déplacés dans le répertoire "xls", etc.

Une fois les fichiers triés, le dossier de destination devra être archivé dans un fichier tar compressé (à l'aide des commandes tar et gzip), dont le nom est soit contenu dans le 2^{ème} paramètre, soit demandé par le script si le 2^{ème} paramètre n'existe pas déjà.

Pour tester votre script, vous pouvez utiliser l'archive "fichiers.tar.gz". Cette archive contient un certain nombre de fichiers avec extensions.

Exercice 3

Créez un nouveau script nommé "auto_namer" dont l'utilité est de renommer automatiquement les fichiers contenus dans un répertoire passé en paramètres. Les noms des fichiers renommés devront être composés d'un préfixe passé en paramètres, d'un underscore ('_'), d'un numéro, incrémenté pour chaque fichier et de l'extension originale du fichier.

Exemple:

```
auto_namer dossier_images prefixe
dossier_images/
  grenouille.GIF
  elephant.jpg
  arbre.png
  banane.JPG
  girafe.jpeg
  mammoth.gif
  guitare.pct
=>
  prefixe_1.GIF
  prefixe_2.jpg
  prefixe_3.png
  prefixe_4.JPG
  prefixe_5.jpeg
  prefixe_6.gif
  prefixe_7.pct
```

Ce script doit permettre de gérer correctement les 2 paramètres (nom du répertoire et préfixe) et informer l'utilisateur si un paramètre manque ou n'est pas valable.

Pour tester votre script, vous pouvez utiliser l'archive "fichiers.tar.gz". Cette archive contient un certain nombre de fichiers avec extensions.