

### Exercice 1 : Entraînements

**Q 1.** Écrire un programme, nommé `compter`, qui lit des mots sur l'entrée standard (un par ligne) et affiche le nombre d'occurrences de chacun.

**Q 2.** Écrire un script, nommé `archive`, qui copie tous les fichiers d'extension `.txt` du répertoire courant en des fichiers de même nom et d'extension `.txt.archive` (toujours dans le répertoire courant).

**Q 3.** Écrire un script, nommé `taille`, qui affiche pour chaque fichier du répertoire courant une ligne de la forme :

```
taille <tabulation> <nom_fichier>
```

**Q 4.** Écrire une commande, nommée `numéroter`, à un argument, un nom de fichier, qui permet de l'afficher sur la sortie standard en préfixant chaque ligne par son numéro. En fonction de la manière dont vous avez écrit la commande, que se passe-t-il si elle est utilisée avec 0, 1 ou 2 arguments ?

**Q 5.** Écrire une commande, nommée `remplacer`, qui permet de saisir au clavier un nom de fichier d'entrée, un nom de fichier de sortie, un motif à remplacer, un motif de remplacement. La commande doit remplir le fichier de sortie avec le résultat de la substitution demandée sur le fichier d'entrée. Ajouter les tests nécessaires (confirmation o/n) pour ne pas écraser le fichier de sortie s'il existe déjà.

### Exercice 2 : Échauffement

Une université permet aux étudiants qui s'inscrivent de choisir une option. Plusieurs services prennent les inscriptions et plusieurs secrétaires se chargent de la saisie avec des outils différents. Le responsable de la formation récupère un fichier texte dont les lignes sont de la forme :

```
nom[sep]prenom[sep]annee[sep]option
```

Les nom, prénom et option peuvent être écrits en toute combinaison de majuscules et minuscules, l'année est forcément 1 ou 2, les séparateurs varient au gré des vents, ils sont composés de virgules, point virgules, espaces et tabulations. Le responsable de la formation –un adepte des bases de données– désire éclater le fichier en trois fichiers *propres* pouvant être transmis à une base de données :

– `f`, contient des lignes de la forme :

```
num;nom;prenom
```

`num` est un numéro d'étudiant (unique) que vous devez attribuer pour chaque étudiant ;

– `f1`, contient des lignes de la forme :

```
num;option
```

qui indiquent le numéro d'étudiant et l'option pour les étudiants de l'année 1 ;

– `f2`, contient des lignes de la forme :

```
num;option
```

qui indiquent le numéro d'étudiant et l'option pour les étudiants de l'année 2.

**Q 1.** Écrire une commande, nommée `ranger`, qui lit sur son entrée standard le fichier reçu par le responsable de la formation et crée les trois fichiers `f`, `f1` et `f2`.

### Exercice 3 : Exercices du DS de 2000-2001

Le but de cet exercice est de rassembler plusieurs fichiers de clientèle provenant de sources différentes. Chaque fichier se présente sous la forme d'une suite de lignes contenant pour chacune un numéro de client, un séparateur qui peut être un nombre quelconque de caractères espace, tabulation, virgule et point virgule et enfin le nom du client. Chaque fichier ayant été saisi séparément des autres, le même numéro peut avoir été attribué dans différents fichiers (nous supposons que tous les noms sont distincts), il faut donc établir une nouvelle numérotation.

**Q 1.** Écrire une commande, nommée `ex3`, qui prend en argument un nombre quelconque de noms de fichier et qui rassemble les informations dans un fichier de sortie au format suivant :

- chaque ligne contient un numéro, une tabulation et le nom du client en majuscules ;
- le nouveau numéro est composé de deux parties séparées par un point. La première partie est l'ancien numéro du client, la seconde partie permet de distinguer les doublons : on attribue au premier client numéro 12 que l'on rencontre le numéro 12.1, au deuxième le numéro 12.2 et ainsi de suite ;
- le nom du fichier de sortie doit être demandé à l'utilisateur (saisi au clavier) ;
- **Indication** Une solution simple est d'utiliser une table de hachage.

**Q 2.** Même exercice que précédemment en ajoutant les points suivants :

- si le fichier de sortie demandé existe déjà, le script doit demander confirmation avant d'écraser le fichier en question ou de demander une nouvelle saisie (jusqu'à ce que le nom de fichier soit correct) ;
- le fichier de sortie doit être trié par ordre de numéros croissants.

#### **Exemple**

```
bash$ cat fichier1
1 dupont
2 durant
3 toto
4 jones
```

```
bash$ cat fichier2
1 germain
12 alphonse
3 tutu
25 tintin
```

```
bash$ ./ex3 fichier1 fichier2
Nom du fichier de sortie : fichier.trie
Voulez-vous ecraser le fichier (O/n) ? n
Nom du fichier de sortie : fichier.txt
```

```
bash$ cat fichier.txt
1.1      DUPONT
1.2      GERMAIN
12.1     ALPHONSE
2.1      DURANT
25.1     TINTIN
3.1      TOTO
3.2      TUTU
4.1      JONES
```