

## Travaux dirigés et pratiques n° 4

### Fichiers

#### Partie TD (2 séances)

##### Exercice 4.1 (Ecriture de fichier)

Soit l'algorithme ci-dessous (déjà vu en cours).

1. Que fait cet algorithme ?
2. Modifier cet algorithme afin qu'il ne stocke que les lettres minuscules et majuscules non accentuées (et aucun autre caractère) dans un fichier nommé "lettres.txt". De plus, à la fin, l'algorithme affichera à l'écran le message stocké.
3. Simuler cet algorithme lorsque l'utilisateur saisit, dans cet ordre : 'a', '+', '4', 'b', '=', 'c' et '.'.

##### lexique

```
flux_fichier flux
chaîne de caracteres nom
caractere c
debut
1  ecrire("fichier ?")
2  lire(nom)
3  ouvrir(flux , nom) en ecriture
4  ecrire ("Donnez un caractere (. pour terminer)")
5  lire(c)
6  tant que non (c = '.' ) faire
7      ecrire(c) dans flux
8      ecrire ("Donnez un caractere (. pour terminer)")
9      lire(c)
10 fin tant que
11 fermer( flux )
fin
```

##### Exercice 4.2 (Lecture de fichier)

Soit l'algorithme ci-dessous (déjà vu en cours).

1. Que fait cet algorithme ?
2. Modifier cet algorithme afin qu'il demande à l'utilisateur le nom d'un fichier d'entiers, affiche tous les entiers positifs contenus dans ce fichier puis indique le pourcentage d'entiers positifs dans le fichier.
3. Simuler le fonctionnement de cet algorithme sur le fichier "gains.txt" = <2; -10; 0; 48; -3>.

```

lexique
    flux_fichier flux
    chaîne de caracteres nom
    caractere temp
debut
1  ecrire("fichier ?")
2  lire(nom)
3  ouvrir(flux , nom) en lecture
4  si (flux) alors
5      lire(temp) dans flux
6      tant que non (fini(flux)) faire
7          ecrire(temp)
8          lire(temp) dans flux
9      fin tant que
10 fermer(flux)
11 sinon
12     ecrire ("ouverture fichier ", nom, " impossible")
13 fin si
fin

```

### Exercice 4.3 (Concaténation de fichiers)

Il s'agit d'écrire un algorithme qui demande à l'utilisateur le nom de deux fichiers contenant des données quelconques, puis les concatène dans un troisième fichier dont le nom est également saisi par l'utilisateur.

### Exercice 4.4 (Calculs)

Écrire un algorithme qui calcule et affiche la moyenne, la valeur minimale et la valeur maximale d'une série d'entiers notés dans un fichier dont le nom est choisi par l'utilisateur.

### Exercice 4.5 (Fusion de fichiers)

Deux fichiers nommés "liste1.txt" et "liste2.txt" contiennent chacun une séquence de mots non accentués triés par ordre alphabétique.

1. Écrire un algorithme qui affiche la liste triée par ordre alphabétique des mots issus de ces deux fichiers.

2. Simuler son fonctionnement pour les fichiers :

"liste1.txt" = <"abeille"; "chameau" "girafe"; "panda">

"liste2.txt" = <"chenille"; "papillon"; "saumon"; "sauterelle">

### Exercice 4.6 (Histogramme)

Écrire un algorithme qui affiche (à l'écran ou dans un fichier) l'histogramme d'une série de notes entières comprises entre 0 et 10 stockées dans un fichier nommé "notes.txt". Chaque barre de l'histogramme est dessinée à l'aide d'étoiles.

1. L'histogramme est horizontal.

2. L'histogramme est vertical.

Exemple (avec des notes allant de 0 à 5) : "notes.txt" = <4; 0; 2; 3; 3; 4; 5; 3>

Histogramme horizontal	Histogramme vertical
0 *	
1	
2 *	*
3 * * *	* *
4 * *	* * * *
5 *	0 1 2 3 4 5

## Partie TP (2 séances)

### 1 Téléchargement de deux programmes

Télécharger les programmes [lecture\\_fichier.cpp](#) et [ecriture\\_fichier.cpp](#) qui sont la transcription des algorithmes présentés en cours. Les étudier, tester leur fonctionnement.

Vous pouvez réutiliser ces programmes pour vous aider à transcrire des algorithmes de la partie TD (question 3).

### 2 Transcription

Transcrire quelques algorithmes de la partie TD de cette feuille en programmes C++.

### 3 Lecture

Télécharger et étudier le document [fluxCpp.pdf](#) pour approfondir vos connaissances sur les fichiers.