

1 Saisie de données

Une association souhaite automatiser son processus d'inscription. Pour cela elle souhaite une application qui demande :

- le nom ;
- le prénom ;
- la date de naissance ;

Ces informations sont affichées à l'écran pour confirmation sous la forme **nom prénom (date de naissance)** et stockées dans un fichier nommé « **adhesion.txt** » sous la forme :

```
1 nom
2 prenom
3 date de naissance
```

1. Écrivez ce programme.
2. Modifiez votre programme de manière qu'on stocke, à chaque exécution du programme, les informations saisies en fin de fichier.

2 Lecture de données et stockage dans un tableau

Créez un fichier "entiers.txt" dans lequel vous écrivez sur chaque ligne et dans l'ordre un entier de 1 à 10.

1. Écrivez une fonction qui lit le fichier ligne par ligne et remplit un tableau par les entiers selon l'ordre indiqué dans le fichier.
2. À partir du tableau, écrivez dans un fichier "decroissant.txt" la liste des entiers par ordre décroissant.

3 Analyse d'un fichier

On souhaite analyser les entrées des musées de La Rochelle. Pour cela on dispose du fichier « **entree-musee-par-annee-origine.txt** ».

La première ligne du fichier contient le nombre d'entrées, la seconde le montant, puis le pays d'origine, la région, le département, le musée et l'année. Et ainsi de suite pour toutes les entrées.

1. Écrivez un programme qui compte le nombre d'entrées.
2. Modifiez votre programme afin de compter le nombre de Québécois.

On souhaite extraire, parmi tous les enregistrements, ceux concernant un musée et un pays donné puis les stocker dans un fichier avec un enregistrement par ligne.

1. Écrivez et testez la fonction réalisant cette extraction. Déclaration à respecter : `void extractMuseum (string inFileName, string outFileName, string museum, string country)` ; Exemple pour un enregistrement de visiteurs suisses au Musée maritime

4 Analyse d'un fichier de log

Tout système d'information (bibliothèque numérique, boutique en ligne, banque, assurance ...) dispose d'un fichier log dans lequel on stocke les traces d'exécution des utilisateurs. Dans cet exercice, on souhaite analyser les actions des utilisateurs suite à leurs navigation dans une bibliothèque numérique. Pour cela on dispose d'un fichier log où chaque ligne représente l'action réalisée par l'utilisateur dans le portail web de la bibliothèque à un moment donné.

Chaque ligne du fichier "Eventlogs.csv" est composée de trois colonnes (séparées par ";"¹). La première colonne contient l'identifiant de l'utilisateur (**case_id**), la seconde décrit l'action réalisée (**activity_name**) par l'utilisateur dont l'identifiant est (**case_id**). La troisième colonne indique l'heure d'exécution de l'action (**timestamp**).

Afin d'extraire la trace d'exécution de chaque utilisateur, nous pouvons regrouper Les lignes qui ont le même **case_id** représente une trace de navigation.

1. D'autres types de séparateurs (délimiteurs) peuvent être utilisés dans ce genre de fichier tels que les tabulations, les virgules, les hashtags...

1. Écrivez une fonction qui ouvre le fichier log en lecture et compte le nombre total d'activités exécutées par l'ensemble des utilisateurs
2. Écrivez une fonction qui calcule et qui retourne la fréquence de l'activité **home_index**. Vous pouvez vous servir de la fonction `find`² de la bibliothèque `string`. Faites la même chose avec d'autres activités (Définir une fonction).
3. Écrivez une fonction qui prend en entrée un `case_id` d'un utilisateur et une activité et qui renvoie le nombre de fois où l'utilisateur a exécuté cette activité.

2. <https://www.techiedelight.com/fr/check-if-a-string-contains-another-string-in-cpp/>