

Feuille de TP - I**Exercice 1** (comptages dans un texte).

On considère un texte stocké dans une string. **Le texte ne comprend que des lettres majuscules ou le caractère “espace” (‘ ’).**

- (1) Écrire une fonction `int mots(string texte)` qui renvoie le nombre de “mots” du texte. Un mot est une séquence maximale de lettres. Deux mots peuvent être séparés par un nombre quelconque d’espaces. Le texte peut débuter ou finir avec des espaces.
- (2) Écrire une fonction `void occurrences(string texte, int t[])` qui stocke dans `t` (supposé de dimension 26) le nombre d’occurrences de chaque lettre dans le texte.
Indication : ramener une lettre majuscule à un indice entre 0 et 25.
- (3) Écrire une fonction `void tri(int nbOcc[], char occTrie[])` qui prend en entrée un tableau `nbOcc` de dimension 26 qui contient le nombre d’occurrences de chaque lettre dans un texte et qui stocke dans le tableau `occTrie`, de dimension 26, les lettres par nombreS d’occurrences décroissantS. S’il y a plusieurs lettres qui ont le même nombre d’occurrences, leur ordre relatif n’importe pas.
- (4) Écrire une fonction `void histogramme(string texte)` qui trace un histogramme des fréquences d’occurrences des lettres dans le texte. Les fréquences d’apparition sont indiquées dans la dernière colonne de l’histogramme, les lettres apparaissent en légende sous l’histogramme. On ne demande pas que la hauteur des colonnes de l’histogramme soit proportionnelle soit à la fréquence d’occurrence.

Exercice 2 (Somme chiffre à chiffre).

On veut effectuer la somme de deux nombres entiers positifs ou nuls qui peuvent avoir une valeur arbitrairement grande qu’on ne peut pas forcément représenter par un des types entier de la machine. Les nombres sont fournis sous la forme de leur représentation en chaînes de caractères. On transmet aussi le résultat sous la forme de sa représentation textuelle.

En-tête : `string somme(string s1, string s2)`

Votre fonction doit vérifier que les deux arguments ne sont pas vides et ne contiennent que des chiffres. Comme les entiers ne sont pas représentables directement par un entier de la machine, on procèdera à la somme chiffre à chiffre en tenant compte de la retenue.

Exemple : somme("10", "991") doit renvoyer la chaîne "1001".

`somme("18446744073709551615", "18446744073709551615")` renvoie "36893488147419103230".

On pourra se servir d’une des versions de la méthode `insert` de `string` dont l’en-tête est `string& insert(size_t pos, size_t n, char c)`

qui insère `n` occurrences du caractère `c` à la position `pos` de la chaîne. Les positions sont comptées à partir de 0. La méthode renvoie une référence sur la chaîne (ce qu’on peut ignorer ici).