

## Sujet 1 : Introduction à C++, Le premier +

### **Exercice 1 : classe `string`, flots `cin` et `cout`, passage par référence. =====**

**Question 1 :** Écrire une fonction qui a pour paramètres une chaîne (`string`) et un entier `d` et qui modifie la chaîne en la chiffrant (ou déchiffrant) en effectuant un décalage de `d` crans sur chaque lettre (de manière circulaire).

Exemples :

- la phrase « Bonjour tout le monde » avec le décalage 6 devient « Hutpuax zuaz rk sutjk »
- la phrase « Ts x'frzxy gnjs !» avec le décalage -5 devient « On s'amuse bien !»

**Question 2 :** Ajouter un `main` qui

- lit une phrase,
- lit un entier
- chiffre la phrase avec l'entier
- affiche la phrase chiffrée

### **Exercice 2 : surcharge de fonctions =====**

**Question 1 :** Écrire une fonction `miroir` qui a pour paramètre une chaîne, et la modifie pour obtenir son image miroir. Par exemple, la chaîne « tout va bien » devient « neib av tuot ».

**Question 2 :** Écrire une fonction de même nom (`miroir`) qui prend en paramètre un entier, et le modifie en le renversant. Par exemple l'entier 1234 devient l'entier 4321.

**Question 3 :** Écrire un `main` de test.

### **Exercice 3 : tableau, vector et list** =====

On considère  $n_j$  joueurs qui forment une ronde, ainsi qu'une comptine de  $n_s$  syllabes. On égrène la comptine en supprimant de la ronde le  $n_s$ -ième joueur. On recommence jusqu'à n'avoir plus qu'un seul joueur. C'est le gagnant.

Par exemple, si on a 10 joueurs et une comptine de 4 syllabes, le jeu se déroule ainsi :

- le joueur 4 est éliminé,
- le joueur 8 est éliminé,
- le joueur 2 est éliminé,
- le joueur 7 est éliminé,
- le joueur 3 est éliminé,
- le joueur 10 est éliminé,
- le joueur 9 est éliminé,
- le joueur 1 est éliminé,
- le joueur 6 est éliminé,
- => le gagnant est le joueur 5 !

**Question 1 :** on stocke les joueurs (numérotés de 1 à  $n_j$ ) dans un tableau de  $n_j$  entiers. Écrire un programme qui simule ce jeu.

**Question 2 :** Même question avec un conteneur `vector` (cf par exemple [cplusplus.com/reference/stl](http://cplusplus.com/reference/stl)). On pourra utiliser `begin`, `iterator`, `end`, `erase`, `push_down`, etc)

**Question 3 :** La suppression (ou l'insertion) d'un élément au milieu d'un `vector` de taille  $n$  est de complexité  $O(n)$ . Par contre, avec le conteneur `list`, elle est de complexité constante. Écrire alors le programme avec une `list`.

### **Exercice 4 : fonction générique** =====

Écrire une fonction générique qui trie aussi bien un tableau contenant des entiers, qu'un tableau contenant des caractères, qu'un tableau contenant des réels, qu'un tableau contenant des chaînes, bref un tableau contenant des éléments pour lesquels l'opérateur `<` est défini.

Tester votre fonction sur différents tableaux.

On n'utilisera ni `vector`, ni `list`, ni ..., mais les bons vieux tableaux du langage C.

L'algorithme de tri à utiliser est celui que vous voulez !