

## Un peu de programmation procédurale avec la STL

### Exercice 1 (palindromes)

Un palindrome est une chaîne de caractères qui se lit de la même manière de gauche à droite et de droite à gauche : otto, madam... Ecrire un programme qui lit une chaîne de caractères sur l'entrée standard, vérifie s'il s'agit d'un palindrome et affiche le résultat.

### Exercice 2 (moyenne)

- Ecrire un programme qui lit une liste de 10 notes sur l'entrée standard et affiche leurs somme et leurs moyenne.
- Sauriez-vous modifier le programme précédent afin que la liste de notes soit lues dans un fichier ?

**Exercice 3 (automate cellulaire de dimension 1)** Ecrire un programme qui affiche l'évolution d'un automate cellulaire booléen de dimension 1 : chaque cellule contient un état booléen et à chaque transition son nouvel état dépend de l'état des cellules voisines. Toutes les cellules changent d'état en même temps. L'état initial sera tiré au hasard. Le nouvel état d'une cellule est calculé en faisant le **ET** logique entre ces deux voisines.

### Exercice 4 (cmpx)

- Ecrire une fonction `cmpx` qui prend en arguments deux entiers et les échange s'ils ne sont pas dans l'ordre croissant.
- Utiliser cette fonction pour écrire une fonction `insert` qui prend en arguments un entier `n` et un vecteur d'entiers triés `vec` et insère `n` dans `vec` au bon endroit.
- Utiliser cette fonction pour écrire un programme qui lit une liste d'entiers sur l'entrée standard et la trie dans l'ordre croissant.
- Sauriez-vous modifier le programme précédent afin qu'il fonctionne non seulement pour les entiers mais pour tous les types où une relation d'ordre est définie ?

### Exercice 5 (bookmaker)

Un bookmaker enregistre dans un fichier chacune de ses transactions sous forme d'une ligne contenant le nom d'un client et une somme positive si le client lui doit de l'argent et négative sinon.

- Définir un nouveau type `struct Bookrecord` permettant de stocker une transaction ie un nom de client et une somme d'argent.
- Ecrire un programme qui lit un fichier de transactions et les enregistre dans un vecteur où chaque élément du vecteur contient une transaction.
- Afficher le contenu de ce vecteur sur la sortie standard.
- Entrer dans une boucle où l'on lit sur l'entrée standard un nom de client et l'on affiche la somme algébrique correspondant aux montants concernant ce client. La boucle se termine sur le caractère fin de fichier.
- Définir les opérateurs d'entrées/sorties pour un `struct Bookrecord`.
- Modifier le programme précédent afin d'utiliser ces nouveaux opérateurs.