

C++

TP n°1

Master Informatique IFI-RIF-MBDS 1ère année
Université de Nice-Sophia Antipolis

1 Pour commencer en C++

Suivez la procédure donnée en TP pour vous procurez les fichiers `test.h`, `test.cpp`, `main.cpp` et le fichier `makefile`. Exécutez la commande `make` dans le répertoire où se trouve le `makefile`. Si tout fonctionne bien, l'exécutable `exec` est créé dans le dossier `bin`. Exécutez le.

2 Tableau

Créez une classe `Array` qui permet de gérer un tableau d'entiers (`int`).

Cette classe a un constructeur par défaut qui alloue 10 éléments, un constructeur qui prend un entier qui alloue le nombre d'éléments spécifiés et un constructeur par copie. La classe possède aussi un destructeur. Ecrivez le code source des ces fonctions dans la classe.

Créez un fichier source et déportez le code dans le fichier source.

La classe dispose également des fonctions membres suivantes

- `void set(const int i, const int val)` qui affecte le i^{eme} élément du tableau avec la valeur `val`.
- `int get(const int i)` qui renvoie la valeur du i^{eme} élément du tableau.
- `int capacity()const` qui renvoie la taille du tableau.

Ecrivez le code C++ de ces fonctions.

Ecrivez une fonction membre `void print()` qui affiche les éléments du tableau.

Ecrivez une fonction membre `nonDecreasingSort()` qui trie les éléments du tableau par ordre croissant.

Testez vos fonctions.

Ecrivez maintenant une fonction membre dont la syntaxe est un peu particulière: `int operator[] (const int i)`.

Cette fonction renvoie le résultat de la fonction `get`. Ecrivez une exemple où vous utilisez le code suivant

```
Array myArray(10);
myArray.set(3,5);
std::cout << "valeur de 3:" << myArray.get(3) << std::endl;
int j=myArray[3];
std::cout << "valeur de j:" << j << std::endl;
```

Cette fonction est un operateur: vous venez de redéfinir l'opérateur `[]`.

Essayez maintenant d'affecter `myArray[3]`. Que se passe t'il ?

Le C++ vous donne un moyen de pouvoir affecter la variable renvoyée par une fonction. On utilise le symbole `'&'` après le type renvoyé. Autrement dit, on écrit: `int& operator[] (const int i)`. Pour que cela fonctionne il ne faut pas appeler la fonction `get` mais directement la valeur du tableau.

Réécrivez votre code de tri.

3 Tableau Extensible

On désire maintenant pouvoir étendre le tableau si besoin. Ecrivez une fonction `void expand(const int n)` qui augmente la taille du tableau. Faites bien attention à gérer la mémoire et à bien recopier les éléments de l'ancien tableau interne dans le nouveau.

4 Pile

Le but de cette section est d'écrire une classe `Stack` définie pour les entiers (`int`). Cette classe implémente une structure de donnée de pile, autrement dit une structure de données de type LIFO (Last In First Out).

On utilisera en interne un tableau dont on augmentera la taille si besoin.

Vous devez en plus des constructeurs par défaut, constructeur par copie et destructeur, définir les fonctions suivantes. Précisez si elles sont `const` ou pas.

- `bool empty()` qui renvoie vrai si la pile est vide et faux sinon.
- `int top()` qui renvoie la valeur au sommet de la pile
- `void pop()` qui dépile la pile
- `void push(const int i)` qui empile l'entier *i*.
- `int size()` qui renvoie la taille de la pile, autrement dit le nombre d'éléments empilés.