

## Programmation C++ STL vector (tableau dynamique)

### 1 définition

La classe **vector** est proche des tableaux en C. Tous les éléments contenus dans le vector sont adjacents en mémoire, ce qui permet d'accéder immédiatement à n'importe quel élément. L'avantage du vector comparé au tableau en C est sa faculté à se réallouer automatiquement en cas de besoin lors d'un **push\_back** ou d'un **resize** par exemple. Le conteneur vector peut être considéré comme un tableau dynamique.

### 2 exemple 1

```
#include <vector>
#include <iostream>

using namespace std;

int main() {
    vector<int> tableau(5,69); // crée un tableau avec 5 fois 69;
    tableau.push_back(4); // ajoute un élément
    tableau.push_back(2);
    tableau.push_back(5);

    // Pour parcourir un vector (même const) on peut utiliser les index
    for(size_t i=0; i<tableau.size(); ++i) {
        cout << tableau[i] << ' ';
    }
    cout << endl;
    tableau.pop_back(); // retire le dernier élément
    tableau.insert(tableau.begin()+1,8); // ajoute 8 en deuxième position
    // Pour parcourir un vector (même const) on peut utiliser les itérateurs
    for (vector<int>::iterator it = tableau.begin(); it != tableau.end(); ++it){
        cout << *it << ' ';
    }
}
```

```
cout << endl;
return 0;
}
```

```
#include <vector>
#include <iostream>

using namespace std;

int main() {
    vector<int> tableau(5,69); // crée un tableau avec 5 fois 69;
    vector<int> image(1);      // deuxième tableau avec un élément

    image = tableau;
    image.resize(20);

    // Pour parcourir un vector (même const) on peut utiliser les itérateurs
    for (vector<int>::iterator it = image.begin(); it != image.end(); ++it){
        cout << *it << ' ';
    }
    cout << endl;
    return 0;
}
```

### 3 Méthodes

push_back(element) :	ajoute un élément
pop_back(element) :	retire un élément
insert(position, element)	insert un élément à la position de l'itérateur
begin()	renvoie l'itérateur du début
end()	renvoie l'itérateur de fin
size()	renvoie le nombre d'éléments
resize()	redéfinit la taille du tableau
front()	renvoie le premier élément
back()	renvoie le dernier élément
empty()	renvoie vrai si vide