

Les vecteurs

Pour pallier les défauts inhérents à la rigidité des tableaux de taille fixe, la librairie (générique) standard de C++ fournit un type de donnée dénommée **vector** (**vecteur**), offrant au programmeur un moyen très efficace pour construire des structures de données permettant de représenter des tableaux de tailles variables (i.e. tableaux dynamiques).

La taille de ces « tableaux » n'est pas obligatoirement prédéfinie, et peut donc varier en cours d'utilisation.

Pour pouvoir utiliser ces vecteurs dans un programme, il faut, comme dans le cas des entrées-sorties, importer les prototypes et définitions contenus dans la librairie, au moyen de la directive d'inclusion: `#include <vector>`

Déclaration et initialisation

```
vector<type> identificateur;
```

```
vector<int> age;
```

```
vector<int> age(5); /*Correspond à la déclaration d'un tableau d'entiers,  
initialement composé de 5 éléments valant 0*/
```

Les fonctions utiles de la classe vector :

- **push_back** : Ajout un élément à la fin du vecteur

```
vector<int> v;  
v.push_back(10); // Ajoute 10 à la fin du vecteur
```

- **pop_back** : Retire le dernier élément du vecteur

```
v.pop_back(); // Retire l'élément 10 de la fin du vecteur
```

- **size** : retourne le nombre d'éléments du vecteur

```
cout << v.size(); // Affiche le nombre d'éléments du vecteur
```

- **erase** : Supprime un élément ou un intervalle d'éléments du vecteur, puis déplace les éléments suivants pour combler l'espace libéré.

```
v.erase(v.begin() + 2); // Supprime le 3ème élément (indice 2) du vecteur  
v.erase(v.begin(), v.begin() + 3); // Supprime les 3 premiers éléments du vecteur
```

- **clear**: vide le vecteur, en supprimant tous ses éléments

```
v.clear(); // Supprime tous les éléments du vecteur
```