

Série 9.2

Exercice 1 : Etudions `vector`

Echauffement avec les vecteurs (tableaux dynamiques)

Pour utiliser le type `vector`, il faut inclure la bibliothèque qui définit ce type avec la directive suivante:

```
#include <vector>
```

Soient les déclarations suivantes:

```
typedef vector<int> MyArray; MyArray tab1, tab2;
```

En vous aidant d'un programme, répondez aux questions suivantes :

A. Expliquer quelles valeurs contiendra `tab1`, après l'exécution de la boucle suivante ?

```
for (int i=0; i<10; i++)  
    tab1.push_back(tab1.size());
```

B. Que fait la fonction suivante ?

```
void foo(MyArray& tab1, MyArray& tab2) {  
    for (int i=0; i < tab1.size(); i++)  
        tab2.push_back(tab1.front());  
}
```

C. Que se passe-t-il lorsqu'on ajoute "trop" d'éléments à un vecteur ? *Faire un programme pour mettre en évidence le mécanisme utilisé par `vector` pour résoudre le problème*

D. Que se passe-t-il si on écrit `tab1[i] = 3;` avec $i \geq \text{tab1.size()}$?

E. Que se passe-t-il si on écrit `tab1.at(i) = 3;` avec $i \geq \text{tab1.size()}$?

Exercice 2 : Comportement dynamique de vector

On demande d'étudier le programme ci-dessous.

- Écrire ce qui va s'afficher sur la console.
- Expliquer le résultat.

```
// comparing size, capacity and max_size
#include <iostream>
#include <vector>
using namespace std;

int main ()
{
    vector<int> myvector;

    // set some content in the vector:
    for (int i=0; i<100; i++)
        myvector.push_back(i);

    cout << "size: " << (int) myvector.size() << "\n";
    cout << "capacity: " << (int) myvector.capacity() << "\n";
    cout << "max_size: " << (int) myvector.max_size() << "\n";
    return 0;
}
```

- Taper le code
- Essayer d'ajouter le maximum d'éléments. Est-ce le comportement attendu ?
- Compléter avec deux façons d'afficher le contenu du vecteur o Avec un itérateur o Avec un *range based for loop*