

Sujet 4 : classe File

Partie 1

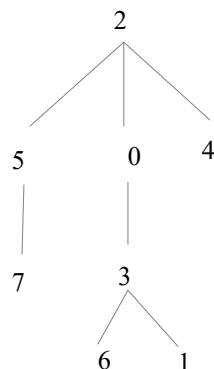
Écrire en C++ une classe permettant de gérer des files pouvant contenir n'importe quel type. On pourra utiliser le conteneur `vector` pour représenter le tableau contenant les éléments de la file.

Partie 2

Tester votre classe sur une file d'entiers, puis sur une file de string.

Partie 3

On représente un arbre ayant n sommets numérotés de 0 à $n-1$ par une classe contenant le nombre de sommets, le numéro de la racine, puis un `vector` de `vector` d'entiers contenant les fils de chaque sommet (le premier élément du `vector` est un `vector` contenant les fils du sommet 0, le deuxième les fils du sommet 1, etc). Par exemple l'arbre



aura pour

- `_nbsommets` la valeur 8,
- `_racine` la valeur 2
- `_fils` le `vector` de `vector` d'entiers :
`{{3}, {}, {5, 0, 4}, {6, 1}, {}, {7}, {}, {}}`

Écrire un programme qui effectue le parcours en largeur d'un tel arbre à l'aide de votre patron de file (sur l'exemple, on doit obtenir : 2 5 0 4 7 3 6 1).
Le tester sur un autre arbre de votre choix.

(On pourra récupérer le fichier `arbre.cpp` qui se trouve sur moodle et le compléter. Ce fichier définit une classe `arbre` qui contient un constructeur ayant en paramètres le nombre de sommets, la racine et les fils, ainsi que l'affichage de l'arbre sous la forme

```
sommet 0 a pour fils : 3  
sommet 1 a pour fils :  
sommet 2 a pour fils : 5 0 4  
...).
```