Buron Baptiste Quentin Macé

8INF840 – Structures de données avancées et leurs algorithmes

Devoir 2

Partie 1

Pour cette première partie, l'arbre choisi est un arbre binaire structuré verticalement dont le but est de pouvoir récupérer les ancêtres d'une personne, voici la structure du nœud :

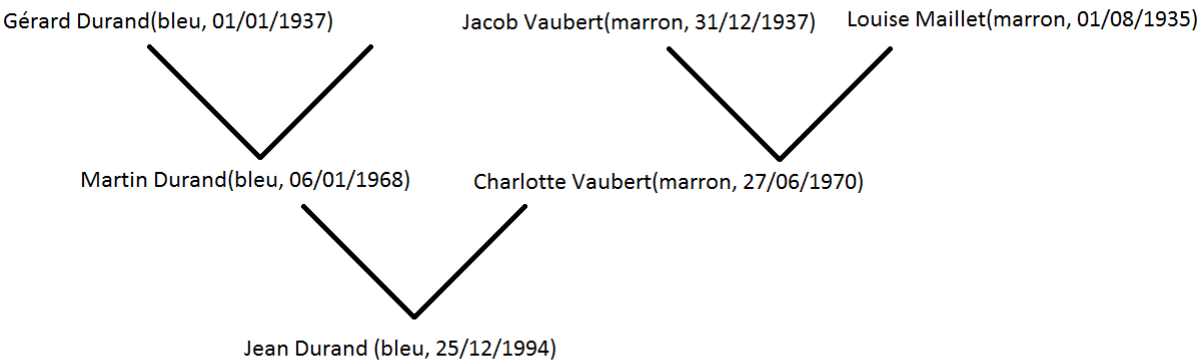
```
class Noeud {
	public:

		T data; //l'élément associé au noeud
		Noeud* pere; //représente en réalité le fils gauche d'un noeud au sein d'un arbre binaire, ici c'est le père pour la cohérence de l'arbre genealogique
		Noeud* mere; //représente en réalité le fils droit d'un noeud au sein d'un arbre binaire, ici c'est la mère pour la cohérence de l'arbre genealogique
		//constructeur du noeud
		Noeud(const T& data_item, Noeud* father = 0, Noeud* mother = 0) : data(data_item), pere(father), mere(mother) {}

//…

}
```

L'arbre accepte les opérations suivantes :

- Ajouter un nœud
- Supprimer un nœud
- Obtenir l'élément d'un nœud
- Parcours d'arbre, 3 modes :
  - Parcours préfixe
  - Parcours postfixe
  - Parcours enfixe
- Lister les personnes selon leurs yeux
- Calculer la moyenne d'âge de la famille

Pour chaque opération on part de la racine de l'arbre puis on effectue le traitement voulu à partir d'un nœud choisi par l'utilisateur.

Pour les besoins du devoir : voici la structure d'une famille générée à l'exécution du programme :



L’utilisateur peut ensuite agir sur l’arbre :