

## Projet 2 Généalogie

3 séances

L'objet du projet est de réaliser un programme de consultation de données généalogiques représentées sous forme arborescente.

### 1 Description des représentations

On va considérer les liens de filiation directs qui existent entre des individus, identifiés par leur sexe (caractère 'f' ou 'm'), leur nom (chaîne de caractères comportant uniquement des lettres) et leur date de naissance (chaîne de caractères au format AAAA-MM-JJ).

#### 1.1 Fichier généalogique

Les individus et leurs liens de filiation directs sont donnés dans un fichier texte au format suivant :

```
Sexe1 Nom1 Date1
...
SexeK NomK DateK
I1 I11 I12
...
In In1 In2
```

où les  $K$  premières lignes décrivent chacune un individu, et les  $n$  lignes suivantes indiquent des liens de filiation : l'individu  $I_j$  a pour père  $I_{j1}$  et pour mère  $I_{j2}$ , ces trois valeurs étant les numéros de ligne des individus concernés dans la première partie du fichier ( $1..K$ ), 0 représentant un parent inconnu. Toutes les données du fichier sont séparées par des tabulations ('\t').

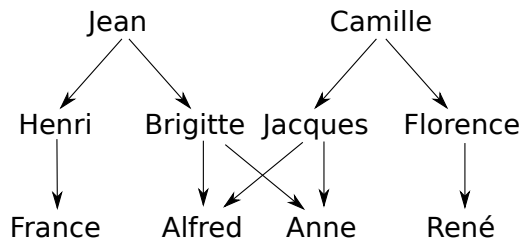
**Exemple 1** *Voici un exemple de fichier généalogique :*

m Jean 2004-02-29	f Florence 1982-09-25
m Henri 1975-04-18	m René 1957-11-16
f France 1951-03-03	1 2 4
f Brigitte 1978-04-21	2 0 3
m Alfred 1954-08-19	4 5 6
f Anne 1953-04-28	7 8 9
f Camille 2007-01-15	8 5 6
m Jacques 1980-06-06	9 10 0

#### 1.2 Arborescences des ancêtres

Chaque individu dispose de 0, 1 ou 2 ancêtres directs (ses parents, s'ils sont connus). Si un individu est associé à un nœud, alors une arborescence binaire permet de représenter les liens de filiations existants entre les ancêtres d'un individu donné. Les arborescences décrites dans un fichier généalogique s'entrelacent pour former un ensemble d'arborescences partageant des nœuds, appelé forêt.

**Exemple 2** *Le schéma ci-dessous représente la forêt des ancêtres décrite dans le fichier de l'exemple 1.*



La structure de données concrète utilisée pour représenter une telle forêt sera constituée d'un tableau dynamique (`std::vector`) de nœuds, chacun contenant un individu et les indices de ses nœuds parents ( $-1$  pour un parent inconnu) ; et d'une table associative implémentée par hachage (`std::unordered_map`) associant à un individu son numéro de nœud dans le tableau.

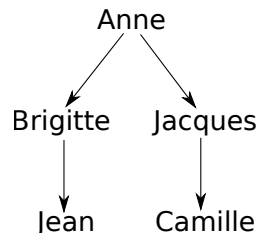
**Exemple 3** La forêt tirée du fichier de l'exemple 1 sera représentée par la structure ci-dessous.

Tableau de nœuds							Table associative		
	individu	père	mère		clé	valeur			
0	m Jean 2004-02-29	1	3		m Jean 2004-02-29	0			
1	m Henri 1975-04-18	-1	2		m Henri 1975-04-18	1			
2	f France 1951-03-03	-1	-1		f France 1951-03-03	2			
3	f Brigitte 1978-04-21	4	5		f Brigitte 1978-04-21	3			
4	m Alfred 1954-08-19	-1	-1		m Alfred 1954-08-19	4			
5	f Anne 1953-04-28	-1	-1		f Anne 1953-04-28	5			
6	f Camille 2007-01-15	7	8		f Camille 2007-01-15	6			
7	m Jacques 1980-06-06	4	5		m Jacques 1980-06-06	7			
8	f Florence 1982-09-25	9	-1		f Florence 1982-09-25	8			
9	m René 1957-11-16	-1	-1		m René 1957-11-16	9			

### 1.3 Arborescence des descendants

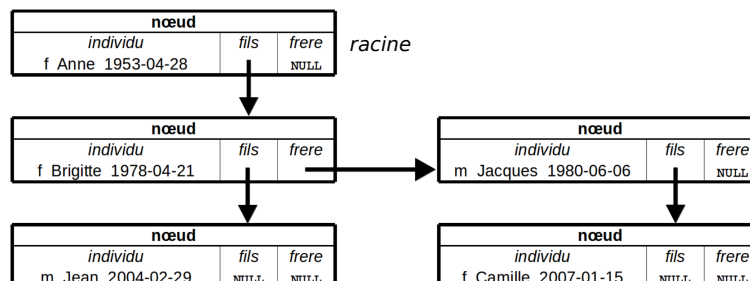
Chaque individu dispose de 0, 1 ou plusieurs descendants directs (ses enfants). Si un individu est associé à un nœud, alors une arborescence  $k$ -aire permet de représenter les liens de filiation entre les descendants d'un individu donné.

**Exemple 4** Le schéma ci-dessous représente une arborescence des descendants tirée du fichier de l'exemple 1.



La structure de données concrète utilisée pour représenter de tels arbres sera constituée de nœuds alloués dynamiquement, accessibles depuis un nœud racine, chaque nœud contenant un individu et des pointeurs vers le nœud représentant son premier enfant et vers un.e frère/sœur suivant.e. Le premier fils sera toujours l'aîné.e de la fratrie ; chaque fratrie sera ordonnée par date de naissance croissante.

**Exemple 5** L'arbre tiré de l'exemple 4 sera représentée par la structure ci-dessous :



## 2 Travail demandé

Le projet devra être rendu avant le **mercredi 3 mai à midi** par dépôt d'une archive ZIP sur Madoc. L'archive sera intitulée aux noms des membres du binôme. Elle comprendra :

- le code source documenté de votre projet ;
- votre rapport (environ 10 pages) au format PDF.

### 2.1 Implémentation

L'implémentation demandée doit s'appuyer sur le squelette fourni sur Madoc dans l'archive **projet2.zip** qui comprend :

- Une définition complète de la classe **Individu** (fichiers **individu.hpp** et **individu.cpp**).
- Une définition de la classe **Ancetres** (fichiers **ancetres.hpp** et **ancetres.cpp**) à compléter.
- Une définition de la classe **Descendants** (fichiers **descendants.hpp** et **descendants.cpp**) à compléter.
- Un programme de tests (fichier **test.cpp**) à compléter.

*Attention : le programme fourni provoque une erreur à l'exécution du fait que les méthodes des classes **Ancetres** et **Descendants** ne sont pas toutes implémentées.*

- Un fichier **makefile** regroupant les commandes de compilation.  
*Pour compiler l'ensemble du projet, il suffit de taper "make" dans un terminal positionné dans le dossier contenant le code.*
- Un fichier **doxyfile** de configuration pour l'outil **doxygen** permettant d'engendrer la documentation du code.
- Un fichier **registre.txt** correspondant à l'exemple 1 pour réaliser des tests.  
*Vous pouvez également proposer d'autres fichiers de tests.*

Le travail demandé consiste à implémenter toutes les méthodes manquantes des classes **Ancetres** et **Descendants**. Pour ce faire vous complétez le code des méthodes dans les fichiers **ancetres.cpp** et **descendants.cpp** sans modifier les fichiers d'entête (**ancetres.hpp** et **descendants.hpp**), si ce n'est pour éventuellement leur ajouter des méthodes privées dont l'utilité sera discutée dans votre rapport, et pour compléter la documentation des méthodes publiques (complexité en particulier). Vous devez également compléter le programme de test, au fur et à mesure de votre progression, afin d'illustrer le bon fonctionnement des méthodes implémentées.

Le code remis devra comprendre l'intégralité des fichiers du projet, à savoir exactement les mêmes que ceux fournis dans l'archive **projet2.zip** sur Madoc.

### 2.2 Rapport

Votre rapport comportera les parties suivantes :

1. Une introduction présentant les objectifs du projet dans le cadre du module (selon vous).
2. Une analyse des structures arborescentes considérées, notamment leur composition et leur complexité spatiale. Vous discuterez en particulier de l'intérêt de la table associative dans la classe **Ancetres**, de l'importance du fait qu'elle soit implémentée par hachage, et de l'utilité qu'elle pourrait avoir (ou ne pas avoir) dans la classe **Descendants**.
3. Une présentation synthétique des algorithmes que vous avez mis en œuvre pour chaque méthode des classes que vous avez implémentées. Vous discuterez en particulier les choix importants que vous avez faits pour leur implémentation, et leur complexité temporelle.
4. Une conclusion résumant l'état final de votre travail, les difficultés rencontrées, et les améliorations envisagées.

La présentation et la rédaction du rapport seront prises en compte dans son évaluation.