

Arbre généalogique

Ce TP consiste à créer et manipuler des arbres généalogiques. Notez que cela implique d'inverser les termes parents / enfants par rapport au vocabulaire classique sur les arbres.

Pensez à tester vos sous-programmes dès que cela est possible.

1. Créer les types **date**, **personne** (nom, prénom, sexe, date de naissance), et **abGen**, un arbre binaire dont les éléments sont composés d'une personne et de deux pointeurs vers ses parents, et représentant un arbre généalogique.
2. Écrire une procédure **saisiePers** permettant la saisie des informations concernant une personne.
3. Écrire une fonction **unique** qui détermine si les deux paramètres de type personne représentent la même personne (même nom, prénom et date de naissance).
4. Écrire une fonction **chercherAdr** qui retourne l'adresse du nœud où se trouve une personne (et NULL si cette personne n'est pas présente dans l'arbre passé en paramètre).
5. Écrire une procédure **ajoutParent** qui ajoute dans un arbre une personne p1 étant un des parents d'une autre personne p2 passée en paramètre. On ne fera rien si p2 n'est pas présent dans l'arbre et si p2 existe mais que l'ancêtre à ajouter est déjà présent.
6. Écrire une procédure **affichePersonne** qui affiche une personne passée en paramètre.
7. Écrire une procédure **afficheTous** qui affiche l'ensemble des personnes présentes contenues dans un arbre généalogique.
8. Écrire une fonction **nbGenerations** retournant le nombre de générations représentées par un abGen.
9. Écrire une fonction **estArbreGen** qui retourne vrai si l'arbre est un arbre généalogique valide, c'est-à-dire où les dates sont bien ordonnées (les parents sont toujours plus âgés que leurs enfants).
10. Écrire une fonction **reunitArbres** permettant de créer l'arbre généalogique d'une personne, à partir des arbres généalogiques de ses deux parents (sans faire de copie des arbres).
Signature : *abGen reunitArbres (abGen pere, abGen mere, personne enfant)*