Examen TD n°2

La durée de l'examen est de 30 minutes, les notes de cours et de TD sont autorisées. La notation prendra en compte la présentation et la clarté des explications (programmes commentés, dessins explicatifs, ...). L'examen sera noté sur 10 points. Les questions sont indépendantes.

L'objectif de cet examen est d'écrire la structure « arbre généalogique » et certaines fonctions associées. On utilisera pour cela les définitions suivantes :

```
typedef struct s_individu individu_t;

struct s_individu {
  char *nom, *prenom;
  individu_t *pere, *mere;
  int nb_enfants;
  individu_t **enfants;
};
```

Le champ `enfants` de cette structure est donc un tableau de longueur `nb_enfants` dont chaque case est un pointeur vers l'enfant en question.

▷ **Question 1.** Écrivez la fonction créant un individu sans lien de parenté. Son prototype est le suivant :
`individu_t *cree_individu(char *nom, char *prenom);`
Indication : il ne faut pas oublier le malloc.

▷ **Question 2.** Ecrivez une fonction ajoutant un enfant à un couple d'individus existants. Son prototype est le suivant :
`individu_t *cree_enfant(char *nom, char *prenom, individu_t *pere,individu_t *mere);`
Indication : Il faut créer le nouvel enfant, puis modifier le pere et la mere pour indiquer dans ces structures l'existance de ce nouvel enfant.
Indication : La modification du père (et de la mère) demande entre autres un `realloc`.

▷ **Question 3 (QUESTION SUBSIDIAIRE).** Écrivez une fonction libérant la mémoire occupée par un individu, en tenant à jour les structures associées (enfants, parents, *etc.*).

▷ **Question 4.** Ecrivez une fonction créant la famille Dupont. Le pere s'appelle Jean, la mère Jacqueline, et ils ont trois enfants Paul, Henri et Charlotte.
Indication : Servez vous des fonctions définies aux questions précédentes.

▷ **Question 5.** Ecrivez les fonctions suivantes, chacune prenant deux individus `ind1` et `ind2` en parametres :
- est_pere_de : renvoi vrai ssi le `ind1` est le pere de `ind2`.
- est_frere_de : renvoi vrai ssi `ind1` et `ind2` ont les mêmes parents.