

TD 5 : arbres tassés

Ecrire les modules permettant de gérer un **tas** de **truc**.

Un **truc** est défini par sa **taille** en cm, son **poids** en grammes, son **âge** en jours. On considère que chaque truc est unique de par sa taille, son poids et son âge (il n'y a pas deux trucs ayant le même poids, ni la même taille ni le même âge).

Dans un premier temps, on considérera un **tas max** de trucs basé sur la taille.

Ce tas sera représenté dans un tableau **tas** de taille **N** et contenant **nb_val** valeurs utiles :

```
#define N ...  
truc tas[N] ;  
int nb_val ;
```

La racine est en position 0, les fils gauche et droit du nœud i sont respectivement en position $2*i+1$ et $2*i+2$ et **nb_val** est le nombre de valeurs présentes dans le tas.

1. Répondez aux questions suivantes :
 - a. Comment savoir si le nœud i a un fils gauche ? un fils droit ? Programmer les primitives correspondantes.
 - b. Comment savoir si le nœud i est une feuille ? Programmer la primitive.
 - c. Comment connaître le père du nœud i ? Programmer la primitive.
2. Écrire une fonction **ajouter** qui prend en paramètre un **truc** et l'ajoute dans le tas en préservant ses propriétés.
3. Écrire une fonction **retirer** qui délivre la plus grande valeur du tas après l'avoir supprimée du tas tout en préservant ses propriétés.
4. En utilisant les deux fonctions précédentes, écrire un programme qui effectue le tri par tas d'un tableau **T** de **trucs** contenant **N** valeurs, dans l'ordre décroissant.
5. Soit l'énumération suivante `typedef enum {MIN, MAX} type_tas;` Et la variable suivante, `type_tas type;`
 - a. Où doit-on déclarer la variable **type** ?
 - b. Modifiez les primitives **ajouter** et **retirer** afin de prendre en compte le type du tas déterminé par la valeur de la variable **type**.
6. Selon le désir de l'utilisateur, le tas max sera construit selon la **taille**, le **poids** ou l'**âge** des **trucs**. Mettez en place les variables nécessaires. Réécrivez la fonction **init_arbre** afin de prendre en compte cette possibilité. Modifiez les primitives impactées par ce changement.