

# Tas, tris et priorités

L'objectif d'implémenter en C une structure de tas.

0. Rappeler les options de compilation C qui permettent d'avoir :

- Des avertissements supplémentaires à la compilation.
- De meilleurs messages d'erreurs à l'exécution.

La variante étoile du TP correspond aux *Bonus* : indiqués dans l'interface des tas.

## A Tas max

Dans ce TP, on implémente des tas `max` ; et non des tas min comme vus en cours. La définition ce à quoi vous vous attendez : arbre binaire complet où l'étiquette d'un noeud est plus grande que celles de ses enfants.

### A.1 Représentation d'un tas

- Rappeler comment représenter un arbre binaire complet à l'aide d'un tableau. Écrire l'arbre correspondant au tableau : 

2	13	14	7	1	8
---	----	----	---	---	---
  - Est-ce un tas ? Proposer un tas-max contenant les mêmes éléments et l'écrire sous forme de tableau.

Dans ce TP, on fait des tas de nombres flottants (dont, pour rappel, le type en C se note `double` ) On définit alors un tas comme suit :

```
12 /** Type d'un tas max */
13 struct tas_s {
14     double* arr; // Le tas
15     int max_size; // La longueur allouée de arr
16     int len;      // Nombre de noeuds du tas
17 };
18 typedef struct tas_s tas;
```



### A.2 Fonctions sur les tas

Cette question vous laisse en autonomie sur le TP :

- Implémenter en C chacune des fonctions de la signature.
- Testez-les **AU FUR ET À MESURE**. Vous ferez vos tests dans un fichier `main.c` distinct de votre librairie de tas.

## B Maintenant, trions !

On souhaite désormais trier un tableau de flottants en C par la méthode du tri par tas.

- Utilisez ce qui précède pour le faire. *Pour le bonus, commencez par remarquer qu'il vous est interdit d'utiliser `tas_vide...`*

## C Pour occuper les plus rapides

On suppose disposer d'un type de *données* (par défaut des `int`, mais on pourrait choisir autre chose) :

```
1 typedef int data;
```

On se donne maintenant le type suivant :

```
1 struct element_s {  
2     double prio;  
3     data   valeur;  
4 };  
5 typedef struct element_s element;
```

5. Modifier l'implémentation des tas pour y ajouter ces deux types, et implémenter une file de priorité max et non un simple tas max.