

TD I41 - 1 Ensemble dynamiquement alloué

On veut réaliser des primitives permettant de gérer des ensembles bornés E d'entiers ($E \subset \mathbb{N}$) en utilisant des structures. Un ensemble est implémenté par une structure de données contenant 2 informations : le cardinal de l'ensemble et le tableau pour contenir les valeurs de cet ensemble. Le rôle du tableau est celui d'un tampon contenant les valeurs de l'ensemble. Par exemple, pour l'ensemble $E = \{1, 2, 10, 20\}$ on a la structure suivante :

- le tableau dynamiquement alloué est

1	2	10	20
---	---	----	----

. Les valeurs significatives dans le tableau sont toujours ordonnées de manière croissante. Les valeurs non significatives contiennent la valeur NONE (NONE = -1).
- le cardinal est 4.

Le tableau contenant les éléments sera dynamiquement alloué en ayant la taille la plus petite qu'il est possible de calculer.

1 Fonctions de Base

1. Déclarer le type `set_t` correspondant à un ensemble et écrire une fonction `set_t * init_set()` qui initialise l'ensemble passé en paramètre à l'ensemble vide (de cardinal = 0 et le pointeur sur le tableau pointe sur NULL).
2. Ecrire la fonction `int cardinal(set_t E)` qui retourne le cardinal de l'ensemble.
3. Ecrire la fonction `int* Alloc(int n)` qui alloue dynamiquement un tableau de taille n et qui l'initialise à NONE. (attention de vérifier votre allocation)
4. Ecrire la fonction `set_t * array2set(int A[N])` qui crée un ensemble à partir d'un ensemble de valeur triée conservé dans un tableau. (Les valeurs peuvent ne pas complètement remplir le tableau. Dans ce cas, le reste correspond à -1).
5. Ecrire la fonction `void print(set_t E)` qui imprime un ensemble.
6. Ecrire la fonction `bool_t is_elt(set_t E, int elt)` qui rend vrai si l'élément `elt` est dans l'ensemble et faux sinon.
7. Ecrire la fonction `int pos(set_t E, int v)` qui retourne :
 - la position du premier élément **plus grand ou égal** à v .
| Exemple : `pos(E, 2)` retourne 1.
 - si l'élément v est plus grand que tous les éléments de l'ensemble, elle retourne le cardinal.
| Exemple : `pos(E, 21)` retourne 4.
8. Ecrire la fonction `set_t * Inter(set_t E1, set_t E2)` qui correspond à l'intersection des ensembles : $E1 \cap E2$.
9. Ecrire la fonction `set_t * Union(set_t E1, set_t E2,)` qui correspond à l'union des ensembles : $E1 \cup E2$.
10. Ecrire la fonction `set_t * Diff(set_t E1, set_t E2)` qui correspond à la différence des ensembles : $E3 = E1 - E2$.
11. Ecrire la fonction `set_t * Delta(set_t E1, set_t E2)` qui correspond à la différence des ensembles : $E3 = E1 \Delta E2 = (E1 \cup E2) - (E1 \cap E2)$.

Vous testerez toutes vos fonctions, les unes après les autres, en écrivant une fonction `main()` utilisant chacune d'elles.