

TD8-y

1 - Ensembles

Nous allons implémenter un type ensemble (une collection non ordonnée d'éléments uniques) en utilisant des listes chaînées implémentées par les structures suivantes :

```
typedef struct _elt{
    int valeur;
    struct _elt *suivant;
} Element;

typedef struct {
    Element *tete;
} Ensemble;
```

Pour les fonctions vous adopterez un style fonctionnel (avec copie profonde).

- Ecrire une fonction qui renvoie un Ensemble vide.
- Ecrire une fonction prenant en paramètre un Ensemble s et renvoyant 1 si s est vide et 0 sinon.
- Ecrire une fonction prenant en paramètre un Ensemble s et un entier e et qui renvoie 1 si e est membre de s.
- Ecrire une fonction prenant en paramètre un Ensemble s et un entier e et qui renvoie un Ensemble constitué des éléments de s et de l'élément e
- Ecrire une fonction prenant en paramètre deux Ensemble s1 et s2 et qui renvoie 1 si l'ensemble s2 est inclu dans s1.
- Ecrire une fonction prenant en paramètre deux Ensemble s1 et s2 et qui renvoie un ensemble égale à l'union des deux ensembles.
- Ecrire une fonction prenant en paramètre deux Ensemble s1 et s2 et qui renvoie un ensemble égale à l'intersection des deux ensembles.
- Ecrire une fonction prenant en paramètre deux Ensemble s1 et s2 et qui renvoie un ensemble égale à la différence des deux ensembles.

2 - Gestionnaire de mémoire

Vous n'êtes pas satisfait du gestionnaire de mémoire (fonctions malloc et free) défini dans stdlib (vous n'avez pas forcément raison) et vous pensez pouvoir faire mieux (vous avez franchement tort).

- Définir les structures nécessaires pour implémenter un gestionnaire de mémoire.
- Ecrire une fonction prenant en paramètre un entier et qui crée un gestionnaire de mémoire de taille max TAILLE_MAX.
- Ecrire une fonction prenant en paramètre un gestionnaire de mémoire et un entier t et qui renvoie un pointeur sur une zone de mémoire libre de taille t.
- Ecrire une fonction prenant en paramètre un pointeur p et un gestionnaire de mémoire et qui libère la zone mémoire pointée par p.

3 - Dictionnaires

Il manque en C une structure de donnée très utile : le dictionnaire (parfois appelée table associative). C'est une structure de donnée qui permet d'indexer ces éléments par des chaînes de caractères. Dans la suite, nous considérerons des dictionnaires composés d'entiers.

- Définir les structures permettant d'implémenter un dictionnaire.
- Ecrire une fonction qui prend un dictionnaire d, une chaîne clef et un entier e et qui modifie d en y ajoutant l'association (clef,e), et qui renvoie 1 si l'opération c'est bien déroulée (et 0 sinon).
- Ecrire une fonction qui prend un dictionnaire d, une chaîne clef et un entier e (passage par référence), qui modifie e par la valeur associée à clef, et qui renvoie 1 si la clef est présente (et 0 sinon).
- Ecrire une fonction qui prend un dictionnaire d et qui renvoie la liste des clefs présentes dans d.