

Projet de programmation 2

Définition d'interface

On souhaite manipuler des graphes orientés dont les sommets sont étiquetés (représentés) par des entiers. Proposer une interface au module *graphe* dont les fonctionnalités sont les suivantes :

- créer un graphe vide, détruire un graphe donné,
- ajouter/supprimer un sommet étiqueté à un graphe donné,
- ajouter/retirer un arc entre deux sommets étiquetés donnés d'un graphe donné,
- créer un sous-graphe à partir d'un graphe donné d'un ensemble de sommets étiquetés donné.

Liste chaînée

On désire manipuler des chaînes de caractères en tant que listes simplement chaînées de caractères.

1. Définir le type *carliste* qui implémente le type liste de caractères.
2. Définir la fonction *bool carliste_est_vide(carliste l)* qui retourne vrai si la chaîne implantée dans *l* est vide ou non.
3. Définir les fonctions de traduction
 - a. *carliste chaine_vers_carliste(char *s)*
 - b. *char *carliste_vers_chaine(carliste l)*.
4. Définir la fonction *carliste carliste_concat(carliste l1, carliste l2)* qui concatène sans dupliquer les listes passées en paramètre. Les carlistes *l1* et *l2* seront considérées comme détruites après l'appel.

Implémentation des exceptions en C

1. Rappeler le prototype et le rôle des deux fonctions nécessaires à l'implémentation du mécanisme d'exception en C.
2. Quelle structure de donnée est-elle nécessaire à toute implémentation du mécanisme d'exception ?