Travaux Diri és de Pro rammation Impérative n°9

Cours d'informatique de Première Année

—Licence MIEF/SPI—

Liste doublement chaînée

Dans ce TD, on va écrire une liste doublement chainée. Si cette structure de donnée prends plus de place en mémoire, on verra qu'elle permet d'écrire rapidement certaines fonctions.

▸ Exercice 1. Liste doublement chainée

– *Soient les structures suivantes :*

```
struct maillon_s{
char * val;
struct maillon_s * suivant;
struct maillon_s * précédent;
};
typedef struct maillon_s maillon_t;

struct liste_s{
struct maillon_s * premier;
struct maillon_s * dernier;
int taille;
};
typedef struct liste_s liste_t;
```

– *Écrire la fonction* `liste_initialiser` *qui crée une liste vide.*
– *Écrire la fonction* `liste_ajouter_debut` *qui prend en entrée un pointeur sur une* `liste_t` *et un entier* x *et ajoute* x *au début de la liste.*
– *Écrire la fonction* `liste_ajouter_fin` *qui prend en entrée un pointeur sur une* `liste_t` *et un entier* x *et ajoute* x *à la fin de la liste.*
– *Écrire la fonction* `liste_extraire_debut` *qui prend en entrée un pointeur sur une* `liste_t` *extrait le maillon qui se trouve au début de la liste et le renvoit.*
– *Écrire la fonction* `liste_extraire_fin` *qui prend en entrée un pointeur sur une* `liste_t` *extrait le maillon qui se trouve au début de la liste et le renvoit.*
– *Écrire la fonction* `liste_tourner` *qui effectue une permutation circulaire de la liste. Exemple :* $1, 2, 3, 4$ *devient* $4, 1, 2, 3$*. Quelle est la complexité de fonction ? Quelle aurait été la complexité si on avait utilisé un tableau ?*

▸ Exercice 2. Li ne de métro

Une ligne de métro est constituée :

– *D'un numéro qui permet de l'identifier.*
– *D'une liste de station.*

Définir la struture qui permet de stocker une ligne de métro. Écrire un programme dans lequel on crée la ligne 3.5 *(3 bis) avec la liste des stations suivante :*

- *Porte des Lilas*
- *Saint-Fargeau*
- *Pelleport*
- *Gambetta*