

TD 5 de Programmation Bas Niveau : Listes chaînées

Aurélien BOSSARD

Pré-requis : chapitres 1 à 9 du cours de Programmation Bas Niveau.

Exercice 1

Soit `liste_point2D` une structure qui encode une liste chaînée de `Point2D` vu dans le TD précédent. Écrivez la structure `liste_point2D` en réutilisant le type `struct point2D` du TD précédent.

Écrivez les fonctions suivantes :

- **`struct liste_point2D * nouvelleListe(Point2D p)`** qui crée une nouvelle liste à partir d'un `Point2D` donné ;
- **`struct liste_point2D * inserer (struct liste_point2D * maliste, Point2D p)`** qui ajoute un nouveau `Point2D p` en début de liste ;
- **`struct liste_point2D * supprimerPremier (struct liste_point2D * maliste)`** qui supprime le premier point de la liste ;
- **`void parcourir (struct liste_point2D * maliste)`** qui affiche tous les points présents dans la liste ;
- **`int * rechercher (struct liste_point2D * maliste, Point2D p)`** qui renvoie un pointeur vers un tableau contenant les indices de tous les points égaux à `p` stockés dans la liste.

Exercice 2

Soit la liste chaînée suivante :

```
1 typedef struct l_etudiants {
2     char * nom;
3     float sem1;
4     float sem2;
5     float moyenne;
6     struct l_etudiants * suiv;
7 } L_etudiants;
```

Écrivez les fonctions suivantes :

1. **`void calculMoy (L_etudiants *l)`** qui calcule et stocke dans le champ `moyenne` de chaque étudiant de `l` la moyenne des champs `sem1` et `sem2` ;
2. **`void afficheNotesEtudiants (L_etudiants * l, char *rech)`** qui parcourt `l` pour afficher les notes de l'étudiant dont le nom est égal à `rech` ;
3. **`void afficheNotesEtudiants (L_etudiants *l, char ** rech, int trech)`** qui affiche les notes des étudiants dont les noms sont dans le tableau de chaînes de caractères `rech` de taille `trech` ;
4. **`L_etudiants * filtre (L_etudiants * l, float seuil)`** qui crée une nouvelle liste chaînée dans laquelle sont copiés tous les étudiants de `l` dont la moyenne est au dessus de `seuil`.