Programmation 2
Feuille de TD-TP n°5
(Pointeurs, structures et tableaux)

Exercice 1 (TD). Tableau d'adresses de structures (TD)

Soit les définitions de structures suivantes.

```
struct uef {
    unsigned coefficient;
    float note;
};

struct releve_BF_N2_info {
    unsigned long identifiant;
    struct uef algebre;
    struct uef programmation;
    struct uef logique;
};

typedef struct releve_BF_N2_info * lien;
```

La structure `releve_BF_N2_info` représente le relevé des trois unités d'enseignement fondamentales (UEF) d'un(e) étudiant(e) du N2 informatique. Le n° de l'étudiant(e) correspond au champ `identifiant`.

Dans cet exercice, on appellera « tableau de relevés » un tableau t dont chaque variable $t[i]$ est de type `lien`. Si $t[i]$ contient l'adresse d'un relevé, on dira que ce relevé est référencé par $t[i]$.

À chaque mention d'un tableau de relevés, vous supposerez que tout l'espace mémoire nécessaire pour stocker les informations contenues dans les relevés référencés par les variables du tableau a été alloué et que tous les champs ont été initialisés.

Question 1. Soit t un tableau de relevés t. Écrivez l'expression permettant d'accéder à la note de logique du relevé référencé pat $t[i]$.

Question 2. Écrivez la définition d'une fonction

```
float moyenne_BF(lien);
```

qui renvoie la moyenne (pondérée par les coefficients) des trois notes d'UEF du relevé dont l'adresse est reçue comme paramètre d'entrée.

Question 3. Écrivez la définition d'une fonction

```
void afficher_moyenne_tab_releves (lien *t, int taille)
```

qui affiche les moyennes pondérées de tous les relevés référencés par un lien du tableau de relevés t, en respectant le format [1]

```
moyenne BF de l'étudiant(e) 12202206 = 12.24
```

1. Toute ressemblance avec le numéro ou la moyenne de bloc des UEF d'une(e) étudiant(e) réellement inscrit(e) à l'université de Paris XIII ne saurait être que fortuite.

Question 4. Écrivez la définition d'une fonction

```
void trier_moyenne_tab_releves (lien *t, int taille)
```

qui réordonne les liens du tableau de relevés t selon l'ordre croissant des moyennes pondérées des relevés. La taille de t est reçue comme deuxième paramètre d'entrée. La fonction doit implémenter un **tri par insertion**.

Question 5. Écrivez la définition d'une fonction

```
unsigned nombre_BF_valide_tab_releves(lien *t, int taille)
```

qui renvoie le nombre d'étudiant(e)s validant le bloc des UEF parmi celles/ceux dont les relevés sont référencés dans le tableau de relevés t. La taille de t est reçue comme deuxième paramètre d'entrée et **les liens de t sont supposés rangés selon l'ordre croissant des moyennes pondérées des relevés**.

Question 6. Écrivez la définition d'une fonction

```
int rechercher_tab_releves(lien *t, int taille, float m_min, float m_max)
```

qui renvoie l'indice d'un lien du tableau de relevés t qui référence un relevé dont la moyenne pondérée est comprise au sens large entre *m_min* et *m_max*. Si t ne référence aucun relevé dont la moyenne vérifie cet encadrement, la fonction renvoie -1. La taille de t est reçue comme deuxième paramètre d'entrée et **les liens de t sont supposés rangés selon l'ordre croissant des moyennes pondérées des relevés**. La fonction doit implémenter une **recherche dichotomique**, effectuant, dans le cas le moins favorable, un nombre de comparaisons de moyennes qui croît linéairement avec le logarithme de la taille du tableau.

Exercice 2 (TP)

Écrivez un programme qui

1. initialise un tableau de 10 relevés (cf. l'exercice 1 ci-dessus) ;
2. calcule et affiche les moyennes dans le bloc des UEF de tous les relevés du tableau ;
3. trie le tableau selon l'ordre croissant des moyennes ;
4. affiche les moyennes de tous les relevés du tableau trié ;
5. calcule et affiche le nombre d'étudiant(e)s qui valident le bloc des UEF ;
6. affiche le résultat de la recherche d'un(e) étudiant(e) ayant une moyenne comprise entre 12 et 14.

Consignes pour l'écriture du programme.

- Pour l'initialisation du tableau de relevés, vous téléchargerez le fichier
  `initialiser_tab_releves_etu.c`
  qui contient la définition (lacunaire) d'une fonction d'allocation mémoire et d'initialisation ; vous la compléterez de sorte que **les notes d'UEF soient initialisées de façon pseudo-aléatoire à des valeurs entières ou demi-entières de l'intervalle** $[6, 20]$.
- Pour l'affichage des moyennes (étapes 2 et 4), le tri du tableau de relevés selon l'ordre croissant des moyennes (étape 3), le calcul du nombre d'étudiant(e)s qui valident le bloc des UEF (étape 5) et la recherche d'un(e) étudiant(e) dont la moyenne dans le bloc des UEF est comprise entre 12 et 14, votre programme appellera respectivement les fonctions des questions 3, 4, 5 et 6 de l'exercice 1 ci-dessus.