

Devoir maison n°1

Les exercices suivants sont à rédiger en C dans un fichier C bien structuré, à rendre par mail avant le 4 novembre à 20h. On pensera à commenter son code. Le code devra être fonctionnel sur des tests simples. En cas d'impossibilité à accéder à un ordinateur, vous pouvez faire l'exercice sur papier et la notation sera adaptée. Le DM est alors à déposer dans mon casier avant le 4 novembre à 20h.

Exercice 1 Maximum

1. Rappeler la fonction `int max(int* tab, int n)` qui permet de calculer le maximum d'un tableau.
2. Écrire une fonction `int deuxieme_max(int* tab, int n)` qui calcule le 2ème maximum d'un tableau.
Par exemple pour le tableau `[1, 2, 1, 8, 8, 4]`, le maximum est 8 et le deuxième maximum est 8 aussi.
3. Écrire une fonction `int ke_max(int* tab, int n, int k)` qui calcule le kème maximum d'un tableau.
Pour l'exemple précédent, le 3ème maximum est 4 et le 5e maximum est 1.

Exercice 2 Carrés magiques

Un carré magique est une matrice carrée de taille $n \neq 0$ dont les éléments sont les entiers de $\llbracket 1; n \rrbracket$, chaque entier étant présent une unique fois. De plus les sommes sur les lignes, les colonnes et les deux diagonales doivent être les mêmes.

Exemple

2 7 6

La matrice $\begin{pmatrix} 9 & 5 & 1 \\ 4 & 3 & 8 \end{pmatrix}$ est un carré magique. Toutes les sommes valent 15.

4 3 8

Écrire une fonction `bool est_carre_magique(int** matrice, int n)` qui prend en entrée un tableau dynamique à deux dimensions carré, alloué par malloc, et sa taille et renvoie `true` s'il s'agit d'un carré magique et `false` sinon.

Écrire au moins un exemple de carré magique 3×3 sur lequel tester votre fonction.

Exercice 3 Rationnels

Dans cet exercice on va représenter les nombres rationnels en C.

On rappelle qu'un nombre rationnel r est un réel qui peut s'écrire $r = \frac{p}{q}$ avec p et q des entiers, q étant non nul.

On va les représenter par un type structure à deux champs nommés `num`, le numérateur, et `den`, le dénominateur.

Le type est défini en C par les deux lignes suivantes :

```
struct rationnel {int num; int den;};  
typedef struct rationnel rat;
```

1. Écrire une fonction `void affiche_rationnel(rat r)` qui affiche un rationnel r sous forme de fraction.
Par exemple, pour `r.num=3` et `r.den=5`, on affichera les trois caractères `3/5`. Pour `r.num=1` et `r.den=2`, on affichera les trois caractères `1/2`.
2. Écrire une fonction `rat multiplie_rationnel(rat r1, rat r2)` qui permet de renvoyer un rationnel qui est le produit de r_1 et r_2 .
3. Écrire une fonction `rat ajoute_rationnel(rat r1, rat r2)` qui permet de renvoyer un rationnel qui est la somme de r_1 et r_2 .
4. Écrire une fonction `float rat_to_float(rat r)` qui transforme un rationnel en le flottant correspondant.
Par exemple, pour `r.num=3` et `r.den=5`, on renverra le flottant `0.6`. Pour `r.num=1` et `r.den=2`, on renverra le flottant `0.5`.