

Série de TPs N°8

Les Types personnalisés « Structures » et les fonctions récursives

Exercice 1

Un point dans le plan est défini par : son nom (*un seul caractère*) et ses coordonnées. Exemple : A(1.5,2).
Un cercle dans le plan est défini par : son nom (*un seul caractère*), son centre et son rayon. Exemple : C(A(3,4.2),8).

- Définir une structure représentant un point dans le plan.
- Définir une structure d'un cercle dans le plan.

Ecrire **un programme C** qui :

- Lit un ensemble de N cercles ($N \leq 25$).
- Lit M points dans le plan ($M \leq 20$).
- Pour chaque point, affiche tous les cercles qui le contiennent.
- Affiche tous les couples de cercles qui n'ont aucune intersection.

Exercice 2

Soit C un nombre complexe défini par sa partie réelle (a) et sa partie imaginaire (b) dont : $C = a + b \cdot i$.

- Définir une structure pour contenir un nombre complexe.
- Ecrire une fonction `somme_complexe(...)` qui retourne la somme de deux nombres complexes.
- Ecrire une fonction `produit_complexe(...)` qui retourne le produit de deux nombres complexes.
- Ecrire une fonction `module_complexe(...)` qui retourne le module d'un nombre complexe.
- Ecrire une fonction `racine_complexe(...)` qui retourne la racine carrée d'un nombre réel.

En utilisant les fonctions précédentes, écrire un **programme C** qui lit un nombre réel quelconque « R » et deux nombres complexes C1 et C2 ($C_1 = a + b \cdot i$; $C_2 = c + d \cdot i$) et qui affiche ensuite :

- La somme de C1 et C2, // $C_1 + C_2 = (a+c) + (b+d) \cdot i$
- Le produit de C1 et C2, // $C_1 \cdot C_2 = (a \cdot c - b \cdot d) + (a \cdot d + b \cdot c) \cdot i$
- Le module de C1 et C2, // $\|C_1\| = \sqrt{a^2 + b^2}$ et $\|C_2\| = \sqrt{c^2 + d^2}$.
- La racine carrée du nombre réel R.

Exercice 3

Ecrire **un programme C** qui utilise une fonction récursive afin de calculer la somme de « n » premiers nombres entiers ($1 + 2 + 3 + \dots + n$).

Exercice 4

Soit TAB un tableau de N entiers ($N \leq 50$).

- Ecrire une procédure récursive permettant de lire les éléments d'un tableau.
- Ecrire une fonction récursive qui permet de déterminer le minimum dans un tableau.
- Ecrire un **programme C** utilisant les deux sous programmes précédents afin de lire un tableau T de M ($M \leq 50$) entiers et de déterminer la valeur minimale dans le tableau T.

Exercice 5 -facultatif-

Ecrire un **programme C** et en utilisant une **fonction itérative** et sa **version récursive** qui permettent de calculer et d'afficher le N^{ième} terme U_N de la suite de FIBONACCI. Cette dernière est donnée par la relation de récurrence :

- $U_1 = 1$ et $U_2 = 1$
- $U_N = U_{N-1} + U_{N-2}$ (pour $N > 2$).

Exercice 6 -facultatif-

Ecrire une procédure qui retourne le quotient et le reste de la division d'un entier p par un entier q .
En utilisant cette procédure, écrire un **programme C** qui lit deux nombres entiers ensuite affiche leur quotient et reste de division.

Exercice 7 -facultatif-

Soit la procédure suivante :

```
void proc(int n)
{
    if (n>0) {
        printf ("%d\t",n) ;
        proc(n-1) ;
        printf ("%d\t",n) ;
    }
}
```

Question : faite la trace de cette procédure pour $n=4$ et dite ce que fait.

Exercice 8 -facultatif-

Soit l'expression suivante :

$$Exp = \sum_{k=1}^n k^3 = 1 + 2^3 + 3^3 + \dots + n^3.$$

Ecrire une **fonction itérative** et sa **version récursive** qui permet de retourner la valeur de Exp .

Ecrire un **programme C** et en utilisant les deux fonctions afin de calculer la valeur de l'expression Exp pour $n=10, 15, 20$ et 25 .