



UNIX - LANGAGE C

PG-108

F. MORANDAT

Filière : Électronique

Année : 1

Semestre : 5

Date de l'examen : 25/01/2013

Durée de l'examen : 2h

E N S E I R B
M A T M E C A
B O R D E A U X

Documents : autorisés non autorisés
 Calculatrice : autorisée non autorisée
 Autre : Attention à la clarté du code et l'écriture.
 Barème indicatif: 4 ; 2 ; 14

1 Fonctions mystères

Que font ces fonctions ? Les réponses doivent faire une phrase maximum.

```
void mystere1(int l, int a[]) {
    for(int i = 0 ; i < l ; i ++)
        if(a[i] < 0)
            a[i] = -a[i];
}
```

```
void mystere2(int l) {
    while(l > 1)
        l /= 2;
    return l;
}
```

```
int mystere3(int i, int j[], int k) {
    int r = 0;
    while( k-- )
        if(j[k] > i)
            r ++;
    return r;
}
```

```
float mystere4(float i) {
    return i * i * 3.14159265359;
}
```

2 Racine ?

Étant donné la structure suivant qui représente les coefficients d'un polynôme du second degré :

```
struct poly_scnd_t { float a, b, c; };
```

1) Écrire la fonction qui permet de saisir ces coefficients.

- 2) Écrire la fonction qui, étant donné un paramètre de la structure `poly_scnd_t`, affiche $f(x) = Ax^2 + Bx + C$ ou `A`, `B`, `C` sont remplacés respectivement les trois coefficients de la structure.
- 3) Écrire la fonction qui, étant donné une structure `poly_scnd_t` et un nombre, retourne *un* si ce dernier est racine du polynôme (à un EPSILON près), *zéro* sinon.
- 4) Écrire la fonction qui appelle les trois fonctions précédentes et affiche `X est racine de f(x) = ...` ou `X n'est pas racine de f(x) = ...`, ou `X` sera remplacé par un nombre saisi par l'utilisateur.

3 Problème

SVP lire jusqu'à la fin avant de commencer. Même s'il y a un lien entre les questions, on peut y répondre de façon indépendante (pourvu que les pré-requis soient explicitement écrits).

On souhaite modéliser et manipuler des triangles dans le plan réel.

- 1) On considèrera dans la suite qu'un triangle est représenté par trois points (sommets).
 - i) Comment représenter un point (`point_t`)? Écrire le code correspondant.
 - ii) Comment représenter le triangle?
- 2) Écrire les fonctions `print_point` et `print_triangle` qui affiche respectivement les coordonnées d'un point et les différentes coordonnées d'un triangle. *Attention* le résultat doit être humainement lisible.
- 3) Écrire les fonctions `scan_point` et `scan_triangle` qui permettent respectivement de saisir au clavier les coordonnées d'un point et d'un triangle.
- 4) Écrire la fonction qui retourne, pour un triangle donné, son périmètre.
- 5) Écrire la fonction qui retourne l'aire d'un triangle.
- 6) Écrire la fonction qui **retourne** textuellement les propriétés d'un triangle, c'est-à-dire *équilatéral*, *isocèle*, *scalène* (tous les côtés ont des longueurs différentes) ou *dégénéré* (un côté de longueur nulle ou un angle plat). *Il est fortement conseillé d'écrire les prédicats associés.*
- 7) Écrire la fonction qui, étant donné un tableau de triangles :
 - i) retourne le nombre de triangles équilatéral qu'il contient.
 - ii) On étendra la fonction à un second paramètre de type tableau. Ce tableau sera rempli par effet de bord pour retourner le nombre de triangles équilatéral (`[0]`), isocèle (`[1]`) et scalène (`[2]`).
 - iii) Pourquoi la fonction doit prendre un second argument et non retourner directement le tableau?
- 8) Écrire une fonction `main` qui permet de saisir une suite de `n` de triangles (`n` sera passé en argument au programme). La suite de votre programme devra afficher le plus d'information possible sur chacun de ces triangles.