

Examen de Langage C

Durée : 0h45

Aucun document autorisé  
Mobiles interdits

- 1. Écrivez un programme qui lit sur l'entrée standard une opération arithmétique formée de 2 entiers et d'un caractère ('+', '-', '\*' ou '/') et qui écrit sur la sortie standard le résultat de son évaluation. Par exemple, si on saisit 18 - 5, le programme écrit 13. Vous utiliserez l'énoncé **switch**. Vous penserez aussi à traiter les cas d'erreur.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>

int main(void) {
    int a, b, res;
    char op;

    // lire les 2 entiers et l'opérateur
    scanf("%d %c %d", &a, &op, &b);
    // évaluer l'opérateur
    switch (op) {
        case '+': res = a+b; break;
        case '-': res = a-b; break;
        case '*': res = a*b; break;
        case '/': assert(b!=0);
                res = a/b; break;
        default : fprintf(stderr, "opérateur %c inconnu\n", op);
                return EXIT_FAILURE;
    }
    // écrire sur la sortie standard le résultat
    printf("%d %c %d = %d\n", a, op, b, res);
    return EXIT_SUCCESS;
}
```

.....  
On représente les coordonnées d'un point  $p$  du plan cartésien par deux réels (double)  $x$  et  $y$  tels que  $p = (x, y)$ . On dispose de la fonction :

**double distance(double x1, double y1, double x2, double y2)**

qui renvoie la distance entre 2 points  $(x_1, y_1)$  et  $(x_2, y_2)$ . On appelle  $\bar{p}$  la distance d'un point  $p = (x, y)$  par rapport à l'origine  $(0, 0)$ .

- 2. Écrivez en C la fonction **compare** qui compare 2 points (4 réels double)  $p_1 = (x_1, y_1)$  et  $p_2 = (x_2, y_2)$  et renvoie :

-1 si  $\bar{p}_1 < \bar{p}_2$   
0 si  $\bar{p}_1 = \bar{p}_2$   
1 si  $\bar{p}_1 > \bar{p}_2$

```
/*
 * Antécédent : (x1,y1) et (x2,y2) 2 points du plan
 * Rôle : renvoie :
 *
 *          -1 si (x1,y1) < (x2,y2)
 *           0 si (x1,y1) = (x2,y2)
 *           1 si (x1,y1) > (x2,y2)
 */
int compare(double x1, double y1, double x2, double y2) {
    double d1 = distance(0, 0, x1, y1);
    double d2 = distance(0, 0, x2, y2);
    if (d1<d2) return -1;
    if (d1>d2) return 1;
    // d1==d2
    return 0;
}
```

- .....  
► 3. Écrivez en C une fonction qui calcule le pgcd de deux entiers naturels positifs. On rappelle que le plus grand commun diviseur de deux entiers naturels positifs est l'entier naturel le plus grand qui les divise tous les deux. Il est tel que :

$$pgcd(a, b) = \begin{cases} pgcd(a-b, b) & \text{si } a > b \\ pgcd(a, b-a) & \text{si } a < b \end{cases}$$

Vous procéderez par de façon itérative par soustractions successives des valeurs  $a$  et  $b$ , jusqu'à ce que  $a$  et  $b$  soient égaux. **Vous donnerez l'invariant et les affirmations nécessaires.**

```
/* Antécédent: a>0 et b>0
 * Rôle : renvoie le pgcd de a et b
 */
int pgcd(int a, int b) {
    while (a != b) {
        // pgcd(a,b) = pgcd(a-b,b) et a>b
        // = pgcd(a,b-a) et a<b
        if (a>b)
            // pgcd(a,b) = pgcd(a-b,b) et a>b
            a -= b;
        else
            // pgcd(a,b) = pgcd(a,b-a) et a<b
            b -= a;
        // pgcd(a,b)==a et a==b
        return a;
    }
}
```