TP 1 : Introduction à C

12 septembre 2023

Exercice 2/3 :

```
#include <stdio.h>

int main(){
    int x = 5;
    int y = 10;
    int z = x + y;
    printf("%d + %d = %d",x,y,z);
}
```

Exercice 4 :

```
#include <stdio.h>

int main(){
    int x;
    int y;

    printf("Valeur de x : ");
    scanf("%d",&x);

    printf("Valeur de y : ");
    scanf("%d",&y);

    int z = x + y;
    printf("%d + %d = %d",x,y,z);
}
```

Exercice 8 :

Ecrire un programme qui demande à l'utilisateur une valeur d'un entier x (précédé du texte "valeur de x : ") et qui affiche "x est positif", "x est nul" ou "x est négatif" en fonction de la valeur de x.

```
#include <stdio.h>

int main(){
    int x;
```

```
    printf("Valeur de x : ");
    scanf("%d",&x);

    if (x>0){
        printf("x est positif");
    }
    else if (x==0){
        printf("x est nul");
    }
    else{
        printf("x est négatif");
    }
}
```

Exercice 9 :

On rappelle que pour un entier n donné, son successeur dans la suite de Syracuse vaut $3n+1$ si n est impair et $n/2$ si n est pair. On rappelle également qu'on calcule le reste dans la division euclidienne de a par b avec $a\%b$. Ecrire un programme qui demande un entier n et affiche son successeur dans la suite de Syracuse.

CORRECTION : on peut déterminer la parité d'un nombre avec son reste dans la division euclidienne par 2. Il suffit donc d'écrire une instruction conditionnelle qui affectera (à la variable initiale ou à la nouvelle variable) la valeur $n/2$ ou $3n+1$ selon que le reste dans la division euclidienne par 2 soit égal à 0 ou non.

```
#include <stdio.h>

int main(){
    int x;
    int y;

    printf("Valeur de x : ");
    scanf("%d",&x);

    if (x%2==0){
        y = x/2;
    }
    else{
        y = 3*x+1;
    }

    printf("le successeur de %d dans la suite de Syracuse est %d",x,y)
}
```

Exercice 10 :

Ecrire un programme qui demande trois réels a, b et c (avec a non-nul) et qui affiche le nombre de racine de la fonction polynomiale définie par $f(x) = ax^2 + bx + c$.

CORRECTION : on sait qu'une fonction polynôme de degré 2 (avec a non-nul) a 0, 1 ou

2 racines selon que le discriminant $b^2 - 4ac$ soit respectivement négatif, nul ou positif. On utilise donc une instruction conditionnelle pour faire la dissociation de cas.

```
#include <stdio.h>

int main(){
    float a;
    float b;
    float c;

    printf("Valeur de a : ");
    scanf("%f",&a);
    printf("Valeur de b : ");
    scanf("%f",&b);
    printf("Valeur de c : ");
    scanf("%f",&c);

    float d = b*b-4*a*c;
    int r;

    if (d<0){
        r = 0;
    }
    else if (d==0){
        r = 1;
    }
    else{
        r = 2;
    }

    printf("le nombre de racines de la fonction polynomiale définie par "
    "f(x) = %f x²+ %f x+ %f est %d",a,b,c,r);
}
```

Exercice 11 :

Ecrire un programme qui demande trois longueurs positives de côtés l_1, l_2 et l_3 (où aucune n'est supérieure à la somme des deux autres), et indique si le triangle dont les longueurs des côtés valent l_1, l_2 et l_3 est un triangle rectangle.

CORRECTION : le Théorème de Pythagore donne la condition nécessaire et suffisante pour qu'un triangle soit rectangle. Selon que la longueur de l'hypoténuse soit l_1, l_2 ou l_3, on a 3 possibilités pour qu'un triangle soit rectangle : $l_1^2 = l_2^2 + l_3^2$, $l_2^2 = l_1^2 + l_3^2$ ou $l_3^2 = l_2^2 + l_1^2$. On va utiliser une disjonction pour exprimer le fait de vérifier l'une de ces trois conditions.

```
#include <stdio.h>

int main(){
    float l1;
    float l2;
```

```
    float l3;

    printf("Valeur de l1 : ");
    scanf("%f",&l1);
    printf("Valeur de l2 : ");
    scanf("%f",&l2);
    printf("Valeur de l3 : ");
    scanf("%f",&l3);

    if (l1*l1 + l2*l2 == l3*l3
    || l1*l1 + l3*l3 == l2*l2
    || l3*l3 + l2*l2 == l1*l1){
        printf("Le triangle est rectangle");
    }
    else{
        printf("le triangle n'est pas rectangle");
    }
}
```

Exercice 12 :

Ecrire un programme qui demande trois longueurs positives de côtés l_1, l_2 et l_3 (où aucune n'est supérieure à la somme des deux autres), et indique si le triangle dont les longueurs des côtés valent l_1, l_2 et l_3 est un triangle rectangle, isocèle, isocèle rectangle ou équilatéral (ou aucun des 4).

CORRECTION : On peut reprendre la première partie du programme précédent, dans laquelle on demande les valeurs des longueurs à l'utilisateur, et changer les instructions conditionnelles à la suite.

PS : On sera amené à exprimer des conditions longues, qui seront par ailleurs des conjonctions de conditions utilisées à d'autres endroits du programme. Pour simplifier l'écriture, on peut stocker dans des variables booléennes les différentes conditions.

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

int main(){
    float l1;
    float l2;
    float l3;

    printf("Valeur de l1 : ");
    scanf("%f",&l1);
    printf("Valeur de l2 : ");
    scanf("%f",&l2);
    printf("Valeur de l3 : ");
    scanf("%f",&l3);
```

```
    int l1 = strtof(strl1);
    int l2 = strtof(strl2);
    int l3 = strtof(strl3);

    bool rect = l1*l1 + l2*l2 == l3*l3
    || l1*l1 + l3*l3 == l2*l2
    || l3*l3 + l2*l2 == l1*l1;

    bool equi = l1 == l2 && l2 == l3;

    bool isoc = (l1 == l2 || l2 == l3 || l1 == l3) && !equi;

    if (equi){
        printf("Le triangle est équilatéral");
    }
    else if (isoc && rect){
        printf("Le triangle est isocèle rectangle");
    }
    else if (rect){
        printf("le triangle est rectangle");
    }
    else if (isoc){
        printf("le triangle est isocèle");
    }
    else{
        printf("le triangle n'est ni équilatéral, ni isocèle, ni rectangle");
    }

}
```

Exercice 13 :

Ecrire un programme qui demande les coordonnées a_1, a_2, b_1, b_2, c_1, c_2 de trois points A, B et C dans le plan et indique si le triangle ABC est un triangle rectangle.

CORRECTION : Le calcul de AB^2, BC^2 et CA^2 se fait à l'aide d'additions, soustractions et multiplications sur les coordonnées des points. On trouve les longueurs AB, BC et CD en prenant les racines carrées de ces valeurs.

Cependant, comme les longueurs sont positives, on peut vérifier les conditions nécessaires directement sur leurs carrés, sans avoir à passer par le calcul exact des longueurs. On se dispensera donc de l'utilisation de la fonction racine carrée, pour plusieurs raisons : premièrement, utiliser la fonction `sqrt` implique d'utiliser la librairie math, et de faire des calculs superflus, ce qui réduit l'efficacité du programme (un calcul de racine carré étant une opération moins élémentaire qu'une addition ou une multiplication).

Deuxièmement, les flottants sont des **approximations** de nombres réels ; il est impossible de les représenter exactement. En particulier, l'utilisation d'opérations aboutissant entre autre à des calculs de valeurs irrationnelles à de grandes chances de faire apparaître des imprécisions, et faire que deux valeurs censées être mathématiquement égales vont être différentes. On peut

par exemple tester l'égalité entre x et z, tous deux censés valoir 2, dans le programme suivant :

```
#include <stdio.h>
#include <math.h>

int main()
{
    float x = 2;
    float y = sqrt(x);
    float z = y*y;
    printf("%d",z==x);
}
```

Rappel mathématique : pour deux points $A(x_A, y_A)$ et $B(x_B, y_B)$, le carré de la longueur AB vaut $(x_B - x_A)^2 + (y_B - y_A)^2$.

```
#include <stdio.h>

int main(){
    float a1;
    float a2;
    float b1;
    float b2;
    float c1;
    float c2;

    printf("Abscisse de A : ");
    scanf("%f",&a1);
    printf("Ordonnée de A : ");
    scanf("%f",&a2);
    printf("Abscisse de B : ");
    scanf("%f",&b1);
    printf("Ordonnée de B : ");
    scanf("%f",&b2);
    printf("Abscisse de C : ");
    scanf("%f",&c1);
    printf("Ordonnée de C : ");
    scanf("%f",&c2);

    l1_carre = (b_1 - a_1)*(b_1 - a_1) + (b_2 - a_2)*(b_2 - a_2);
    l2_carre = (c_1 - a_1)*(c_1 - a_1) + (c_2 - a_2)*(c_2 - a_2);
    l2_carre = (b_1 - c_1)*(b_1 - c_1) + (b_2 - c_2)*(b_2 - c_2);

    if (l1_carre + l2_carre == l3_carre
    || l1_carre + l3_carre == l2_carre
    || l3_carre + l2_carre == l1_carre){
        printf("Le triangle est rectangle");
    }
    else{
```

```
        printf("le triangle n'est pas rectangle");
    }
}
```