TD2

EXERCICE 1: Etoiles

Écrire un programme qui affiche à l'écran 10 étoiles sous la forme suivante:

```
         *
        *
       *
      *
```

etc....

********** Correction Exercice 1 **********

```
void exercice1() {
  int hauteur = 10;
  int largeur = hauteur;

  while (hauteur > 0) {
    largeur = hauteur;
    while (largeur > 0) {
      printf(" ");
      largeur--;
    }
    printf("*\n");
    hauteur--;
  }
}
```

EXERCICE 2: Table de multiplication

Ecrire un programme qui affiche la table de multiplication des entiers inférieur ou égaux à 12. Votre affichage doit être le suivant:

```
       1   2   3   4   5   6   7   8   9  10  11  12
-------------------------------------------------------
 1 :   1   2   3   4   5   6   7   8   9  10  11  12
 2 :   2   4   6   8  10  12  14  16  18  20  22  24
 3 :   3   6   9  12  15  18  21  24  27  30  33  36
 4 :   4   8  12  16  20  24  28  32  36  40  44  48
 5 :   5  10  15  20  25  30  35  40  45  50  55  60
 6 :   6  12  18  24  30  36  42  48  54  60  66  72
 7 :   7  14  21  28  35  42  49  56  63  70  77  84
 8 :   8  16  24  32  40  48  56  64  72  80  88  96
 9 :   9  18  27  36  45  54  63  72  81  90  99 108
10 :  10  20  30  40  50  60  70  80  90 100 110 120
11 :  11  22  33  44  55  66  77  88  99 110 121 132
12 :  12  24  36  48  60  72  84  96 108 120 132 144
```

********** Correction Exercice 2 **********

```
void exercice2() {
  int i = 1;
  int j;
  int valeur;

  while (i <= 12) {
    printf("%d", i);
    j = 1;
    while (j <= 12){
      valeur = i*j;
      printf("\t%d",valeur);
      j++;
    }
    printf("\n");
    i++;
  }
}
```

EXERCICE 3: Nombres premiers

Écrire un programme qui demande un nombre à l'utilisateur et qui teste si ce nombre est premier ou pas.

********** Correction Exercice 3 **********

```
void exercice3() {
  int nombre;
  printf("Entrez un nombre: ");
  scanf("%d",&nombre);

  int i = 2;
  int res = 1; //Il est premier jusqu'à preuve du contraire
  while (i < nombre) {
    if (nombre%i == 0) res = 0;
    i++;
  }
  if (res == 1) printf("Ce nombre est premier\n");
  else printf("Ce nombre n'est pas un nombre premier\n");
}
```

EXERCICE 4: Nombres amis

Soit n et m, deux entiers positifs. n et m sont dits *amis* si la somme de tous les diviseurs de n (sauf n lui-même) est égale à m et si la somme de tous les diviseurs de m (sauf m lui-même) est égale à n.

Écrire une fonction qui demande à l'utilisateur deux entiers n et m et qui affiche si n et m sont ou non des nombres amis.

Écrire une fonction qui demande à l'utilisateur un entier positif $nmax$ et qui affiche tous les couples de nombres amis (n, m) tels que $n \leq m \leq nmax$.

********** Correction Exercice 4 **********

```
void exercice4_1() {

  int n;
  int m;
  printf("Entrez n: ");
  scanf("%d",&n);
  printf("Entrez m: ");
  scanf("%d",&m);

  /* VOIR L'INTERET DES FONCTIONS POUR EVITER DE REPETER */
  // Calcul de la somme des diviseurs de n
  int i = 1;
  int sommeDivN = 0;
  while (i < n) {
    if (n%i == 0) sommeDivN+=i;
    i++;
  }

  // Calcul de la somme des diviseurs de m
  int j = 1;
  int sommeDivM = 0;
  while (j < m) {
    if (m%j == 0) sommeDivM+=j;
    j++;
  }

  if (sommeDivM == n && sommeDivN == m) printf("Ces nombres sont amis\n");
  else printf("Ces nombres ne sont pas amis\n");
}

void exercice4_2() {

  int n;
  int mMax;
  printf("Entrez n: ");
  scanf("%d",&n);
  printf("Entrez mMax: ");
  scanf("%d",&mMax);

  // Calcul de la somme des diviseurs de n
  int i = 1;
```

```
  int sommeDivN = 0;
  while (i < n) {
    if (n%i == 0) sommeDivN+=i;
    i++;
  }

  // Vérification des amis entre n et mMax
  int k = n;
  while (k <= mMax) {

    // Calcul de la somme des diviseurs de m (k)
    int j = 1;
    int sommeDivM = 0;
    while (j < k) {
      if (k%j == 0) sommeDivM+=j;
      j++;
    }

    if (sommeDivM == n && sommeDivN == k)
      printf("Les nombres %d et %d sont amis\n",n,k);
    k++;
  }
}
```

EXERCICE 5: Racines réelles d'un polynôme du second degré

Ecrivez un programme qui calcule les racines réelles d'un polynôme du second degré. Vous afficherez la valeur exacte des racines, ainsi que leurs valeurs approchées.

********** Correction Exercice 5 **********

```
void exercice5() {

  // Polynome ax2 + bx +c
  int a;
  int b;
  int c = 5;
  printf("Entrez la valeur de a: ");
  scanf("%d",&a);

  printf("Entrez la valeur de b: ");
  scanf("%d",&b);

  printf("Entrez la valeur de c: ");
  scanf("%d",&c);
```

```
  int delta = b*b - 4*a*c;

  // Racines réelles
  float x1;
  float x2;


  if (delta < 0) printf("Il n'existe pas de racines réelles\n");
  else if (delta == 0) {
    x1 = -b/(2*a);
    printf("Il existe une racine réelle: %f\n",x1);
  }
  else {
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
    printf("Il existe deux racines réelles: %f et %f\n",x1,x2);
  }
}
```

EXERCICE 6: Sommes ...

Ecrivez trois version d'un programme qui calcule la série $S(n) = \sum_{i=1}^{n} \frac{1}{i}$. La première version utilisera une boucle *for*, la deuxième une boucle *while*, et la troisième une boucle *do while*. Ecrivez également *une* version qui fait la somme de n à 1 : $S(n) = \sum_{i=n}^{1} \frac{1}{i}$. Comparez les résultats des deux méthodes !

********** Correction Exercice 6 **********

```
void exercice6() {

  int n;
  printf("Entrez n: ");
  scanf("%d",&n);

  /*
   * SOMME DE 1 A N
   */
  float somme = 0;
  // Boucle for
  float i;
  for (i=1 ; i<=n ; i++) {
    somme+=(1/i);
  }
  printf("La somme avec la boucle for de 1 à N est: %f\n",somme);

  // Boucle while
  i = 1;
```

```
somme = 0;
while (i <= n) {
  somme+=(1/i);
  i++;
}
printf("La somme avec la boucle while de 1 à N est: %f\n",somme);


// Boucle do while
i = 1;
somme = 0;
do {
  somme+=(1/i);
  i++;
}
while (i <= n);
printf("La somme avec la boucle do while de 1 à N est: %f\n",somme);

/*
 * SOMME DE N A 1
 */

// Boucle for
i = 1;
somme = 0;
for(i=n ; i>=1 ; i--) {
  somme+=(1/i);
}
printf("La somme avec la boucle for de N à 1 est: %f\n",somme);

// Boucle while
i = n;
somme = 0;
while (i >= 1) {
  somme+=(1/i);
  i--;
}
printf("La somme avec la boucle while de N à 1 est: %f\n",somme);

// Boucle do while
i = n;
somme = 0;
do {
  somme+=(1/i);
  i--;
}
while(i>=1);
```

```
  printf("La somme avec la boucle do while de N à 1 est: %f\n",somme);
}
```

`On voit que les résultats entre les 2 formes ne sont pas identiques.`

EXERCICE 7: Calcul de suite

Calculer les valeurs successives de la suite :

$$u_n = \sqrt{1 + \sqrt{2 + \sqrt{\ldots + \sqrt{n}}}}, \text{ pour } 1 \leq n \leq N.$$

********** Correction Exercice 7 **********

```
  int n;
  printf("Entrez n: ");
  scanf("%d",&n);
  float res = sqrt(n);
  int i=n;
  while (i >= 1) {
    res = sqrt(i+res);
    i--;
  }
  printf("Le résultat est: %f\n",res);
```

EXERCICE 8: Les suites de Syracuse

On se propose de construire un petit programme qui permet d'étudier les suites dites de Syracuse :

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

La conjecture de Syracuse dit que quelle que soit la valeur de départ, la suite finit par boucler sur les valeurs 4,2,1,4,2,1,...

1. Construire un programme qui demande une valeur de départ u_0 et affiche les valeurs successives jusqu'à tomber sur la valeur 1 ;
2. modifier le programme pour qu'il compte les itérations, sans affichage intermédiaire ;
3. modifier le programme pour qu'il redemande éventuellement une nouvelle valeur de départ, à l'aide d'une boucle `do ... while`.

********** Correction Exercice 8 **********

```
void exercice8_1() {

  int u0;
```

```
  printf("Entrez u0: ");
  scanf("%d",&u0);

  int u = u0;
  printf("%d\n",u);

  while (u != 1) {
    if (u%2 == 0) u = u/2;
    else u = 3*u+1;
    printf("%d\n",u);
  }
}

void exercice8_2() {

  int u0;
  printf("Entrez u0: ");
  scanf("%d",&u0);

  int u = u0;
  int i = 1; // Nbr d'itérations

  while (u != 1) {
    i++;
    if (u%2 == 0) u = u/2;
    else u = 3*u+1;
  }

    printf("Nombre d'itérations: %d\n",i);
}

void exercice8_3() {

  int u0;
  printf("Entrez u0: ");
  scanf("%d",&u0);

  int u = u0;
  int i = 1; // Nbr d'itérations

  do {
    if (u == 1) {
      printf("Entré une valeur différente de 1!!!!\n");
      scanf("%d",&u);
    }
    i++;
    if (u%2 == 0) u = u/2;
```

```
    else u = 3*u+1;
  }
  while (u != 1);
  printf("Nombre d'itérations: %d\n",i);
}
```