03.Programmes Python

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Un *programme* est la traduction d'un algorithme dans un langage de programmation.

1 Instructions conditionnelles

Exemple 1. Calcul du maximum de deux nombres : algorithme et programme en Python

Entrée: variables réelles x et y
Sortie: variable réelle M dont la valeur est égale au maximum des valeurs de x et de y
Début
  Lire x, y
  Si $x < y$ **alors**
    $M \leftarrow y$
  sinon
    $M \leftarrow x$
  FinSi
  Afficher M
Fin

```
#Entrée : x,y flottants
#Sortie : maximum M de x et y
x=float(input("valeur de x?"))
y=float(input("valeur de y?"))
if x<y:
    M=y
else:
    M=x
print("le maximum de x et y est :",M)
```

Exercice 1. Enregistrer ce programme dans un fichier nommé *maximum.py*, puis l'exécuter et le tester. Créer puis tester un programme permettant d'obtenir les valeurs approchées des solutions réelles d'une équation du second degré.

2 Boucle Pour

La fonction *range* permet de générer une liste d'entiers.

```
>>> range(1,5)
[1, 2, 3, 4]
```

Exemple 2. Calcul de $1+2+\cdots+n$: algorithme et programme en Python

Entrée: variable entière non nulle n
Sortie: variable entière s dont la valeur est égale à $1+2+\cdots+n$
Début
  Lire n
  $s \leftarrow 0$
  Pour k **allant de** 1 **à** n **faire**
    $s \leftarrow s+k$
  FinPour
  Afficher s
Fin

```
#Entrée : n entier
#Sortie : somme s des entiers de 1 à n
n=int(input("valeur de n?"))
s=0
for k in range(1,n+1):
    s=s+k
print("la somme des entiers de 1 à ",n," vaut ",s)
```

Exercice 2. Enregistrer ce programme dans un fichier puis l'exécuter et le tester. Créer puis tester un programme permettant d'afficher les entiers de 1 à n dans l'ordre décroissant.

3 Boucle Tant Que

Exemple 3. Calcul de la plus petite puissance de deux supérieure ou égale à n : algorithme et programme en Python

Entrée: variable entière n
Sortie: variable entière p dont la valeur est égale à la plus petite puissance de deux supérieure ou égale à n
Début
  Lire n
  $p \leftarrow 1$
  TantQue $p < n$ **faire**
    $p \leftarrow 2p$
  FinTantQue
  Afficher p
Fin

```
# Entrée : n entier
# Sortie : p plus petite puissance de deux supérieure ou égale à n
n=int(input("valeur de n?"))
p=1
while p<n:
    p=2*p
print("la plus petite puissance de deux supérieure ou égale à ",n," est ",p)
```

Exercice 3. Enregistrer ce programme dans un fichier puis l'exécuter et le tester. Créer puis tester un programme permettant d'afficher les multiples de 7 inférieurs ou égaux à un entier n donné.

Exercice 4. Créer un programme permettant à l'utilisateur de tenter de deviner un nombre entier compris entre 1 et 6 choisi au hasard par l'ordinateur.

Réponses

1)
```
#Entrée : a,b,c flottants
#Sortie : x0,x1,x2 valeurs approchées des solutions réelles de l'équation ax^2+bx+c=0
from math import*
a=float(input("valeur de a?"))
b=float(input("valeur de b?"))
c=float(input("valeur de c?"))
d=b**2-4*a*c
if d<0:
    print("l'équation ax^2+bx+c=0 n'admet pas de solution réelle")
else:
    if d==0:
        x0=-b/(2*a)
        print("l'équation ax^2+bx+c=0 admet une unique solution réelle",x0)
    else:
        x1=(-b-sqrt(d))/(2*a)
        x2=(-b+sqrt(d))/(2*a)
        print("l'équation ax^2+bx+c=0 admet deux solutions réelles",x1,"et",x2)
```

2)
```
#Entrée : n entier non nul
#Sortie : entiers i de 1 à n dans l'ordre décroissant
n=int(input("valeur de n?"))
i=n
for k in range(1,n+1):
    print(i)
    i=i-1
```

3)
```
# Entrée : n entier
# Sortie : multiples m de 7 inférieurs ou égaux à n
n=int(input("valeur de n?"))
m=0
while m<=n:
    print(m)
    m=m+7
```

4)
```
from random import*
n=randint(1,6)
print("vous devez tenter de deviner un nombre entier compris entre 1 et 6")
r=0
while r!=n:
    r=int(input("valeur proposée?"))
print("vous avez deviné!")
```