SNT - TP 3 : Les boucles

Dans le TP précédent nous avons vu la notion de fonction qui permet de simplifier des programmes. Cependant, dans certains cas, les mêmes instructions étaient répétées plusieurs fois. Dans ce TP, nous allons voir comment utiliser les boucles pour éviter de répéter les instructions.

Important : Avant de commencer le TP, ne pas oublier de :

- Créer un dossier TP 3 dans votre dossier SNT (qui se trouve sur votre clé usb) ;
- Sauvegarder tous les programmes liés aux exercices suivants dans ce dossier.
- **ATTENTION :** bien lire les consignes concernant le rendu des programmes !

EXERCICE 1

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog1.py » (sans les guillemets !!!!!!!!!!) dans votre dossier TP3.

1. Recopier le programme suivant sur l'éditeur puis l'exécuter.
   (**Attention à bien respecter les tabulations et à ne pas oublier la dernière instruction (mainloop()) !)**

```
from turtle import *

for k in range(4) :
    forward(100)
    left(45)

mainloop()
```

2. À quoi sert l'instruction « for k in range(4) : » ?
3. Modifier le programme pour obtenir un polygone régulier à huit côtés.
   Comment appelle-t-on ce polygone ?

EXERCICE 2

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog2.py ».

1. Recopier le programme suivant dans l'éditeur et l'exécuter :

```
from turtle import *

circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)
circle(50)
left(36)

mainloop()
```

2. À l'aide d'une boucle for, modifier le programme précédent pour obtenir le même résultat.

EXERCICE 3

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog3.py ».

1. Recopier le programme suivant dans l'éditeur et l'exécuter.

```
a=0
for k in range(5) :
    a=a+1
print(a)
```

2. Modifier le programme précédent pour qu'il affiche dans la console les entiers de 1 à 5.
3. Modifier le programme précédent pour qu'il affiche dans la console les entiers de 10 à 50.

EXERCICE 4

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog4.py ».

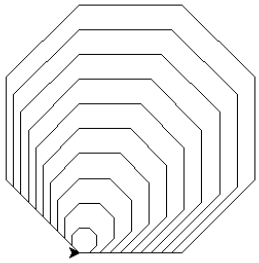
1. Écrire une fonction qui prend en paramètre un nombre c et qui trace un polygone régulier à huit côtés, avec c pour longueur. Le programme doit ressembler à :

```
from turtle import *

def octog(c) :
    à compléter
    à compléter

octog(100)
mainloop()
```

2. Modifier le programme précédent (**sans modifier la fonction octog**) pour afficher le dessin suivant :

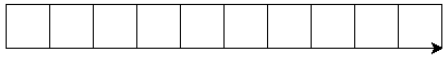


Indication : utiliser une variable pour la longueur qui change à chaque fois.

EXERCICE 5

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog5.py ».

1. En utilisant une boucle pour, écrire une fonction carre qui prend un argument un nombre c et qui trace un carré de côté c.
2. Compléter le programme pour afficher le dessin suivant :



EXERCICE 6

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog6.py ».

1. Recopier le programme suivant dans l'éditeur et l'exécuter.

```
a=1
for k in range(5) :
    a=a+1
print(a)
```

2. Modifier le programme précédent pour afficher les carrés de 1 à 10.
3. Modifier le programme précédent pour qu'il affiche :

   Le carré de 1 est 1.
   Le carré de 2 est 4.
   Le carré de 3 est 9.
   ....
   Le carré de 10 est 100.

EXERCICE 7

Créer un nouveau module python et sauvegarder ce fichier sous le nom « prog7.py ».

1. Recopier le programme suivant dans l'éditeur et l'exécuter.

```
a = 0
S = 0
for k in range(5) :
    a=a+1
    S=S+a
print(S)
```

Quelle valeur obtient-on ? À quel calcul cela correspond-il ?

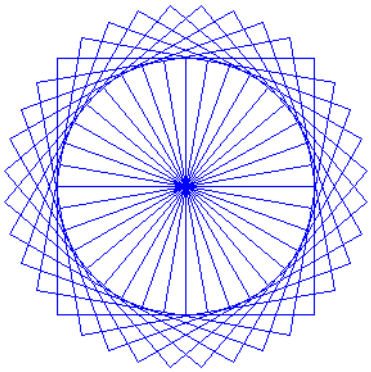
2. Modifier le programme précédent pour obtenir le résultat de la somme suivante :

   $1+2+3+4+5+\cdots+96+97+98+99+100$.

3. Écrire une fonction qui prend en argument un nombre n et qui renvoie le résultat de la somme $1+2+3+4+\cdots+n$.

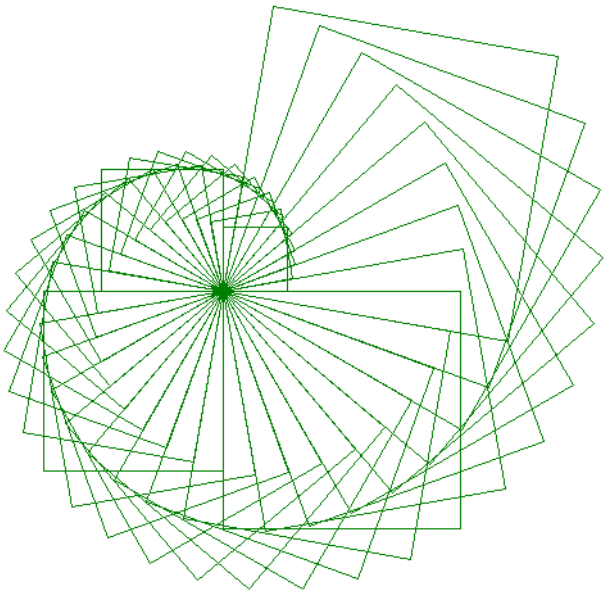
EXERCICE 8

1. Écrire un programme nommé « prog81.py » qui affiche le dessin suivant :



   Indication : on pourra réutiliser la fonction carré définie dans l'exercice 4.

2. Écrire un programme nommé « prog82.py » qui affiche le dessin suivant :



EXERCICE 9

Écrire un programme, que l'on appellera prog9.py, contenant une fonction nommée table et qui prend en argument un paramètre n et qui affiche la table de multiplication de n.