Informatique - TP 3

Utilisation des boucles `for`

M. Marmorat, M. Morel

16 octobre 2024

Exercice 1 **Un premier test** Voici deux exemples d'utilisation de la boucle `for`. Analyser les programmes et prévoir la sortie affichée à l'écran. Les tester.

```
for k in range(1,10):
     print(k)
```

```
for lettre in "Bonjour":
    print(lettre)
```

Exercice 2 **La fonction `range`**

Q1 Écrire un programme affichant tous les nombres entiers entre 0 et 50 (inclus).

Q2 Écrire un programme affichant tous les nombres pairs entre 0 et 50 (inclus).

Q3 Écrire un programme affichant tous les nombres impairs entre 101 et 121 (inclus).

Q4 Écrire un programme affichant tous les nombres multiples de 5 de 20 à -20 (inclus !). On respectera bien ici le sens de l'affichage souhaité : on part de 20 pour aller à -20.

Exercice 3 **Les multiples de 7** Ecrire un programme qui affiche tous les multiples de 7 entre 0 et 70.

Exercice 4 **La fonction factorielle** On rappelle que la factorielle de n, notée $n!$, vaut par définition

$$n! = 1 \times 2 \times \cdots \times (n-1) \times n.$$

Q1 Que valent $5!$, $6!$ et $7!$?

Q2 Écrire une fonction `factorielle` qui calcule la factorielle de n.

Q3 Vérifiez votre fonction en l'appelant pour différentes valeurs de l'argument n.

Exercice 5 **Calcul d'une somme** Écrire une fonction `somme1` prend en argument un entier n et qui calcule

$$\sum_{k=0}^{n} \sqrt{2k+1} = \sqrt{3} + \sqrt{5} + \sqrt{7} + \cdots + \sqrt{2n+1}.$$

On vérifiera que pour $n = 100$, cette somme vaut environ 957.06.

Exercice 6 **Au service des suites récurrentes et des sommes** On considère la suite (u_n) définie par $u_0 = 1$ et

$$\forall n \in \mathbb{N}, \quad u_{n+1} = 3\sqrt{u_n} + 1.$$

Q1 Écrire une fonction `suite3` qui prend en argument un entier n et qui renvoie u_n (testez sur de petites valeurs de n).

Q2 Écrire une fonction somme, utilisant la fonction `suite3`, qui prend en argument un entier n et qui renvoie $\sum_{k=0}^{n} u_k$. A nouveau, testez sur de petites valeurs de n.

Q3 Combien de fois Python calcule-t-il u_1 pour calculer la valeur `somme(5)` ?

Q4 Écrire une fonction `somme_bis` qui prend en argument un entier n et qui renvoie $\sum_{k=0}^{n} u_k$, sans le défaut évoqué à la question précédente.

Exercice 7 Écrire une fonction qui prend en argument un mot (une chaîne de caractères) et qui renvoie le nombre de voyelles contenues dans ce mot. Testez cette fonction sur le mot « informatique ».

Exercice 8 **Au hasard !** Écrire une fonction prend en argument un nombre entier N et qui renvoie la moyenne de N nombres tirés au hasard entre 0 et 100.

Remarque 1. Pour tirer un nombre au hasard, on pourra utiliser la fonction `randrange` do module `random` : le sciprt suivant

```python
import random
print(random.randrange(3, 9))
```

affiche un entier pris uniformément entre 3 et 9.

Exercice 9 **Évolution d'une population de grues du Canada** Dans cet exercice, on modélise l'évolution d'une population de grues du Canada à partir de la population initiale et d'un taux de croissance.

Q1 Si n désigne une année en particulier, p_n la population l'année n et p_{n+1} la population l'année $n+1$, et si r désigne le taux de croissance, quelle relation lie p_n, p_{n+1} et r ?

Q2 Écrire une fonction population prenant en argument la population initiale de grues, un taux de croissance par années, et le nombre n d'années considérées. Cette fonction renverra le nombre de grues après n années d'évolution selon le taux de croissance choisi.

Q3 Vérifier que pour une population initiale de 425 grues et une croissance de 1.94% par an, la population de grues au bout de 30 ans est de 756 individus.

Exercice 10 **Spirale de Fermat des graines de tournesol** On peut modéliser l'implantation des fleurs de tournesol (c'est-à-dire des "graines" rassemblées au cœur de la fleur principale) par une spirale de Fermat. La n-ème fleur (ou graine) se situe alors à une distance d_n du centre de l'inflorescence et forme un angle θ_n avec l'horizontale avec :

$$d_n = \sqrt{n} \quad \text{et} \quad \theta_n = \pi(3 - \sqrt{5})n$$

. On peut alors exprimer les coordonnées de la n-ème fleur x_n et y_n par

$$x_n = d_n \times \cos(\theta_n) \quad \text{et} \quad y_n = d_n \times \sin(\theta_n).$$

Q1 On propose le squelette de script suivant :

```
# on importe le module de trace de graphiques et les fonctions
    mathematiques
import matplotlib.pyplot as plt
from math import sqrt, pi, cos, sin

N =
x = [] #x est la liste des abscisses des graines, on cree la liste vide
y = [] #y est la liste des ordonnees des graines, on cree la liste vide

for n in range(N):
    d_n =
    theta_n =
    x_n =
    y_n =
    x.append(x_n) #on ajoute x_n a la liste x
    y.append(y_n) #on ajoute y a la liste y
```

Complétez les trous dans le script ci-dessus pour calculer les coordonnées de 1000 fleurs de tournesol.

Q2 Complétez le script précédent avec les deux instructions suivantes

```
plt.plot(x,y, "oy", markeredgecolor='gray')
plt.show()
```

pour obtenir la figure suivante

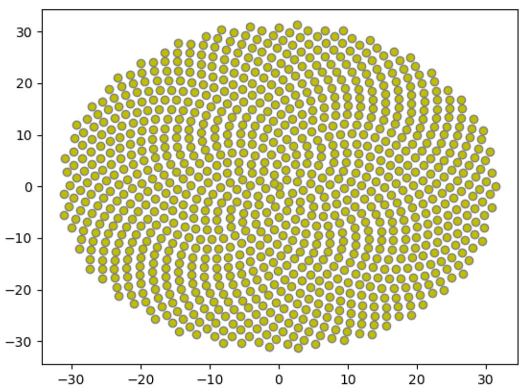

Figure 1 – Une spirale de Fermat à 1000 points !