TP 4 : Boucles ; Introduction aux listes

Exercice 1. Algorithme d'Euclide

On rappelle l'algorithme d'Euclide par division successives permettant de calculer le PGCD de deux nombres `a` et `b` :

entrées : a,b entiers avec a>b>0

$x \leftarrow a$
$y \leftarrow b$
tant que $y \neq 0$ faire :
  $r \leftarrow$ reste de la division euclidienne de x par y
  $x \leftarrow y$
  $y \leftarrow r$
fin tant que

sortie : x

(1) Tester ce programme à la main pour déterminer le PGCD de 24 et 60 puis le PGCD de 222111333 et de 74000.

(2) Implémenter ce programme en Python. On écrira une fonction `Euclide(a,b)`.

(3) Adapter cette fonction pour qu'elle marche encore avec $0 < a < b$.

Exercice 2. Méthode de Monte-Carlo

On considère le programme suivant :

```
import random as rd

def MonteCarlo(n):
    compteur = 0
    for i in range(n):
        x=rd.random()
        y=rd.random()
        if x**2+y**2 <1 :
            compteur = compteur + 1
    return compteur/n
```

(1) Que fait cette fonction ?

(2) Adapter cette fonction de telle sorte à ce qu'elle affiche une valeur approchée de π

(3) Définir une fonction `erreur_pi(estimation)` qui prend en argument une estimation de π et renvoie l'erreur relative commise sur cette estimation.

(4) Tester cette fonction avec l'évaluation de π obtenue à l'aide de la méthode de Monte-Carlo avec $n = 100$, $n = 1000$, $n = 10^4$, $n = 10^5$, $n = 10^6$ et $n = 10^7$

(5) Que conclut-on ?

Exercice 3. Intermède sur les listes

(1) **Premiers exemples**

(a) Taper dans la console `liste=[2,3,5,7,11,13,17,19,21]`. Puis taper `liste[1]`. Comment accéder au $4^{\text{ème}}$ élément de `liste` ?

(b) Modifier le $9^{\text{ème}}$ élément de `liste` en 23.

(c) Taper dans la console `len(liste)`.
Que vaut `liste[len(liste)-1]` ? Et `liste[len(liste)]` ?

(2) **Parcours de listes**

(a) Écrire une fonction `affiche(L)` qui prend en argument une liste `L` et qui affiche tous ses éléments à l'écran.

(b) Écrire une fonction `somme(L)` qui prend en argument une liste `L` de nombres et qui renvoie la somme des éléments de `L`.

(3) **Construction de listes par compréhension**

(a) Taper `L=[i**2 for i in range(10)]` à la console. Que se passe-t-il ?

(b) Générer de manière similaire la liste `X=[0, 0.01, 0.02, ... , 0.99, 1]`

(c) Taper ensuite `Y=[a**2 for a in X]` à la console. Que vient-on de générer ?

Exercice 4. Rendu de monnaie : algorithme glouton

La société Sharp commercialise des caisses automatiques utilisées par exemple dans des boulangeries. Le client glisse directement des pièces ou des billets dans la machine, qui se charge de rendre directement la monnaie.

Afin de satisfaire les clients, on cherche à déterminer un algorithme qui va rendre le moins de monnaie possible.

La machine dispose de billets de 20€, de 10€, et de 5€.

Elle dispose de pièces de 2€, de 1 €, de 50c, 20c, 10c, 5c, 2c et 1c.

On se propose donc de concevoir un algorithme qui demande à l'utilisateur du programme la somme totale à payer, ainsi que le montant inséré par l'acheteur. Le programme affichera alors à l'écran les billets et pièces à rendre par la machine.

Exercice 5. Nombres premiers

Écrire une fonction `premier(n)` qui prend en argument un entier n et renvoie la liste des nombres premiers inférieurs ou égaux à n.

On pourra effectuer une recherche sur internet à propose du crible d'Eratosthène.