
Programmation en Python I – TE n° 730A

Pour chaque problème donné, compléter le fichier Python remis en annexe.

Problème 1 (4 points)

Dans le fichier `probleme_1.py`, écrire une fonction `div_10` qui retourne `True` si un entier est divisible par 10 mais pas par 3 et qui retourne `False` dans le cas contraire.

Tester ensuite cette fonction avec les nombres 40, 60 et 2019 et afficher le résultat dans le `shell` de Python.

Problème 2 (4 points)

Dans le fichier `probleme_2.py` :

- écrire un programme qui demande à l'utilisateur d'entrer deux nombres entiers positifs a et b ;
- le programme effectue ensuite la division euclidienne de a par b en déterminant les deux entiers naturels q et r tels que $a = b \cdot q + r$ et $0 \leq r < b$;
- le programme affiche finalement à la console l'égalité fondamentale de la division de a par b , par exemple pour $a = 100$, $b = 17$, on affiche dans le `shell` de Python $100 = 5 \cdot 17 + 15$.

Par exemple, le programme affiche :

```
Entrez le dividende : 100
```

```
Entrez le diviseur : 17
```

```
L'égalité fondamentale : 100 = 5 * 17 + 15
```

Problème 3 (3 points)

Dans le fichier `probleme_3.py` écrire un programme qui calcule un diviseur propre, c'est-à-dire différent de 1 et de lui-même, du nombre $N=4294967297$.

Problème 4 (5 points)

Dans le fichier `probleme_4.py` :

- créer la variable `PI` qui contient le nombre π importé depuis le module `math` ;
- écrire une fonction `leibniz` qui prend un paramètre entier `n` et qui renvoie une approximation de π basée sur la formule ci-dessous :

$$\pi = 2 \cdot \left(1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \frac{1 \cdot 2 \cdot 3 \cdot 4}{3 \cdot 5 \cdot 7 \cdot 9} + \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}{3 \cdot 5 \cdot 7 \cdot 9 \cdot 11} + \dots \right)$$

le paramètre nommé `n` désigne le nombre de termes à inclure dans la somme ;

- faire afficher dans le `shell` de Python la différence `PI-leibniz(100)`.

Problème 5 (Monte-Carlo — 7 points)

- a) Ouvrir le fichier `probleme_5.py`, l'exécuter et observer le résultat.
- b) Écrire un programme qui fait se déplacer une tortue aléatoirement sur l'écran, dans un carré de 400 pixels de côté, dont le coin inférieur gauche est placé à $(-200; -200)$ et le coin supérieur droit est placé à $(200; 200)$.

Le programme marque un point à l'écran à la fin de chaque déplacement. Le point sera de couleur verte si l'emplacement est dans le disque de rayon 200 pixels centré à l'origine et de couleur rouge sinon.

Le nombre de points est égal à 10000.

- c) A l'aide du nombre de points de couleur verte et du nombre de points de couleur rouge, calculer une approximation du nombre π .

