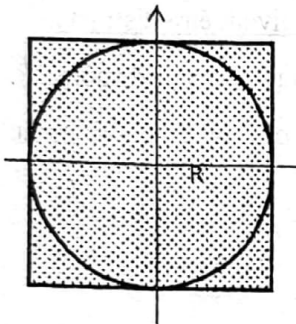


Examen – S2 – 2024/2025

<b>Filière : 1<sup>ère</sup> LFI</b>	<b>Matière : Programmation Python</b>		<b>Enseignants : Dr. Abir Ben Hmida Sakly</b>
<b>Date : 20 / 05 / 2025</b>	<b>Nbr de Crédits : 2</b>	<b>Coefficient : 2</b>	<b>Documents autorisés : Non</b>
<b>Durée de l'examen : 1h30</b>	<b>Régime d'évaluation : Mixte</b>		<b>Nombre de pages : 03</b>
	<b>EX (60%) + DS (25%) + TP (15%)</b>		

Exercice 1 :



Une des méthodes pour approximer la valeur de  $\pi$  consiste à remplir au hasard un carré de côté  $2R$  (où  $R$  est le rayon du cercle) par  $N$  points. Parmi ces points, il y aura  $P$  points qui se trouveront à l'intérieur du cercle inscrit dans le carré. En déterminant  $P$ , on pourra calculer une valeur approchée de  $\pi$ , car le rapport entre le nombre  $P$  de points à l'intérieur du cercle et le nombre total  $N$  de points est une estimation du rapport entre la surface du cercle et celle du carré. Nous avons donc :

$$\frac{P}{N} \approx \frac{\text{aire du cercle}}{\text{aire du carré}} \approx \frac{\pi \times R^2}{4 \times R^2} \approx \frac{\pi}{4} \quad \text{ce qui donne} \quad \pi \approx \left(\frac{P}{N}\right) \times 4$$

1) Chaque point sera représenté par un dictionnaire avec les clés 'x' et 'y' (coordonnées flottantes).

1.1 Ecrire une fonction **generer\_point** qui accepte en paramètre un réel  $R$  et retourne un dictionnaire présentant le point généré. Vous pouvez utiliser la fonction `uniform(a,b)` de la bibliothèque `random` qui renvoie un nombre flottant  $N$  entre  $a$  et  $b$  ( $a \leq N \leq b$ ).

python

```
def generer_point(R):
    """Génère aléatoirement un point dans le carré [-R,R]x[-R,R]
    Retourne un dictionnaire {'x': coord_x, 'y': coord_y}
    """
```

1.2 Ecrire une fonction **est\_dans\_carre** qui accepte en paramètre un point et un réel  $R$  et retourne 1 si le point appartient au carré sinon 0.

Un point appartient au carré c'est-à-dire  $(-R \leq x \leq R)$  et  $(-R \leq y \leq R)$

python

```
def est_dans_carre(point, R):
    """Vérifie si le point est dans le carré de côté 2R
    Retourne 1 si vrai, 0 sinon
    """
```

1.3 Une fonction **est\_dans\_cercle** qui accepte en paramètre un point et un réel  $R$  et retourne 1 si le point se trouve à l'intérieur du cercle de rayon  $R$  sinon 0. Un point  $M$  de coordonnées  $(x,y)$  est à l'intérieur de la sphère si  $\sqrt{x^2 + y^2} \leq R$

python

```
def est_dans_cercle(point, R):
    """Vérifie si le point est dans le cercle de rayon R
    Retourne 1 si vrai, 0 sinon
    """
```

2) L'ensemble des points seront stockés dans une liste POINTS.

2.1 Ecrire une fonction ajout qui permet d'ajouter des points (dont les coordonnées sont générées aléatoirement dans les bornes du carré) dans la liste POINTS. Les points doivent être distincts.

2.2 Ecrire une fonction nbr\_carre qui retourne le nombre de points appartenant au carré de côté  $2R$ .

2.3 Ecrire une fonction nbr\_cercle qui retourne le nombre de points se trouvant à l'intérieur du cercle de rayon  $R$ .

3) Écrire un programme Python qui permet de :

- Saisir un entier  $N$  strictement positif.
- Saisir le rayon  $R$  du cercle (strictement positif).
- Remplir la liste POINTS par  $N$  points distincts générés aléatoirement dans le carré.
- Calculer et afficher la valeur approximative de  $\pi$ .

## Exercice 2 :

On veut créer un système de manipulation de nombres rationnels représentés sous forme de dictionnaires, avec les opérations de base. Chaque nombre rationnel sera représenté par un dictionnaire avec deux clés :

- numérateur (entier)
- dénominateur (entier strictement positif)

Exemple :

```
r1 = {'numérateur': 3, 'dénominateur': 4} # Représente 3/4
```

## Ecrire les fonctions suivantes :

1. Création et simplification d'un rationnel

```
def creer_rational(numer, denom):
    """
    Crée un rationnel simplifié.
    Si le dénominateur est négatif, le signe est reporté au numérateur.
    Lève une erreur si le dénominateur est zéro.
    """
```

## 2. Operations de base

### 2.1. Additionner deux rationnels

### 2.2. Soustraire deux rationnels

### 2.3. Multiplier deux rationnels

### 2.4. Diviser deux rationnels

python

```
def additionner(r1, r2):
    """Retourne la somme de deux rationnels sous forme simplifiée"""

def soustraire(r1, r2):
    """Retourne la différence r1 - r2 sous forme simplifiée"""

def multiplier(r1, r2):
    """Retourne le produit de deux rationnels sous forme simplifiée"""

def diviser(r1, r2):
    """Retourne le quotient r1 / r2 sous forme simplifiée"""
    # Lève une erreur si r2 est nul
```

## 3. Comparer deux rationnels

```
def sont_egaux(r1, r2):
    """Retourne True si les rationnels représentent la même valeur"""
```

## 4. Afficher un rationnel

```
def afficher(r):
    """Retourne une chaîne représentant le rationnel sous forme 'a/b'"""
```

## 5. Ecrire le programme principal permettant de tester toutes les fonctions

```
Entrez le premier rationnel :
Numérateur : 3
Dénominateur : 4

Entrez le deuxième rationnel :
Numérateur : 1
Dénominateur : 2

Résultats :
3/4 + 1/2 = 5/4
3/4 - 1/2 = 1/4
3/4 * 1/2 = 3/8
3/4 / 1/2 = 3/2
Les rationnels ne sont pas égaux.
```

**MAY THE FORCE OF INDENTATION BE WITH YOU, PYTHONISTA ! 🍄**

PS : When life throws errors, remember - there's no **Ctrl+Z**.. but there's always the **#trial\_and\_error** method! 😊