



Sujets A.P.M.E.P

1. Comprendre un programme

1.1 Mars 2023 – Centres Étrangers groupe 1, sujet 1 – Exercice 1.4

- boucle `while`
- test `if ... then ... else`
- fonction `abs`

Soit deux réels a et b avec $a < b$.

On considère une fonction f définie, continue, strictement croissante sur l'intervalle $[a; b]$ et qui s'annule en un réel α .

Parmi les propositions suivantes, la fonction en langage Python qui permet de donner une valeur approchée de α à 0,001 est :

programme 1

```
1 def racine(a, b):
2     while abs(b - a) >= 0.
3         ↵001:
4         m = (a + b) / 2
5         if f(m) < 0 :
6             b = m
7         else:
8             a = m
9     return m
```

programme 2

```
1 def racine(a, b):
2     m = (a + b) / 2
3     while abs(b - a) <= 0.
4         ↵001:
5         if f(m) < 0 :
6             a = m
7         else:
8             b = m
9     return m
```



programme 3

```
1 def racine(a, b):
2     m = (a + b) / 2
3     while abs(b - a) >= 0.
4         ↵001:
5         if f(m) < 0 :
6             a = m
7         else:
8             b = m
9     return m
```

programme 4

```
1 def racine(a, b):
2     while abs(b - a) >= 0.
3         ↵001:
4             m = (a + b) / 2
5             if f(m) < 0:
6                 a = m
7             else:
8                 b = m
9     return m
```

1.2 Mars 2023 – Centres Étrangers groupe 1, sujet 2 – Exercice 1.5

- boucle for

On pose $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{100}$.

Parmi les scripts Python ci-dessous, celui qui permet de calculer la somme S est :

programme 1

```
1 def somme_a():
2     S = 0
3     for k in range(100):
4         S = 1 / (k+1)
5     return S
```

programme 2

```
1 def somme_b():
2     S = 0
3     for k in range(100):
4         S = S + 1 / (k+1)
5     return S
```



programme 3

```
1 def somme_c():
2     k = 0
3     while S < 100:
4         S = S + 1 / (k+1)
5     return S
```

programme 4

```
1 def somme_d():
2     k = 0
3     while k < 100:
4         S = S + 1 / (k+1)
5     return S
```

1.3 Mars 2023 – Polynésie, sujet 1 – Exercice 3.4

- boucle for
- utilisation des listes

Pour chacune des affirmations suivantes, indiquer si elle est vraie ou fausse. Chaque réponse doit être justifiée.

On considère la fonction `mystere` définie ci-dessous qui prend une liste `L` de nombres en paramètre.

On rappelle que `len(L)` représente la longueur de la liste `L`.

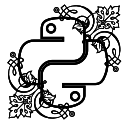
```
1 def mystere(L):
2     S = 0
3     for i in range(len(L)):
4         S = S + L[i]
5     return S / len(L)
```

Affirmation : L'exécution de `mystere([1, 9, 9, 5, 0, 3, 6, 12, 0, 5])` renvoie 50.

Remarques :

- la fonction `mean` se trouve dans les bibliothèques `numpy` et `statistics`
- `dir(nom_du_module_importe)` affiche la liste des fonctions du module.
- `help(nom_de_la_fonction)` affiche l'aide de la fonction.

```
1 import numpy as np
```



```
2 import statistics as st
3
4 dir(statistics)
5 help(st.mean)
6 help(np.mean)
```

2. Comprendre et trouver la valeur renvoyée

2.1 Mai 2022 – Métropole, sujet 1 – Exercice 3

- boucle for
- fonction non usuelle!

2. c) Le programme ci-dessous, écrit en langage Python, utilise la fonction `binomiale(i, n, p)` créée pour l'occasion qui renvoie la valeur de la probabilité $P(X = i)$ dans le cas où la variable aléatoire X suit une loi binomiale de paramètres n et p .

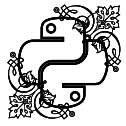
```
1 def proba(k):
2     P = 0
3     for i in range(0, k+1) :
4         P = P + binomiale(i, 20, 0.25)
5     return P
```

Déterminer, à 10^{-3} près, la valeur renvoyée par ce programme lorsque l'on saisit `proba(5)` dans la console Python.

Interpréter cette valeur dans le contexte de l'exercice.

Remarque : La bibliothèque `scipy` contient plusieurs lois de probabilité rangées dans le module `stats`.

```
1 from scipy import stats
2
```



```
3 X = stats.binom(20, 0.25)
4 print(X.cdf(5)) # p(X <= 5)
5 # print(X.pmf(5)) # p(X = 5)
```

2.2 Mars 2023 – Polynésie, sujet 2 – Exercice 4.5

- boucle for
- utilisation des listes
- commentaire
- test if ... then

On considère la fonction `mystere` définie ci-dessous qui prend une liste `L` de nombres en paramètre.

On rappelle que `len(L)` renvoie la longueur, c'est-à-dire le nombre d'éléments de la liste `L`.

```
1 def mystere(L):
2     M = L[0]
3     # on initialise M avec le premier élément de la liste L
4     for i in range(len(L)):
5         if L[i] > M:
6             M = L[i]
7     return M
```

Affirmation : L'exécution de `mystere([2, 3, 7, 0, 6, 3, 2, 0, 5])` renvoie 7.



2.3 Mars 2023 – Métropole, Antilles-Guyane, Maroc, sujet 1 – Exercice 3.A.4

- boucle `while`

On considère le programme ci-contre, écrit en langage Python.

Déterminer la valeur renvoyée par la saisie de `seuil(8.5)` et l'interpréter dans le contexte de l'exercice.

```
1 def seuil(p):
2     n = 1
3     u = 3
4     while u <= p:
5         n = n + 1
6         u = 0.9 * u + 1.3
7     return n
```

3. Programmes à compléter

3.1 Mars 2023 – Asie, sujet 2 – Exercice 3.5

- boucle `for`
- notation `**`
- méthodes sur les listes : `.append`

On modélise la situation à l'aide d'une suite $(v_n)_{n \in \mathbb{N}}$; on note v_0 le nombre de noyaux contenus dans le polonium au début de l'expérience.

Pour $n \geq 1$, v_n désigne le nombre de noyaux contenus dans le polonium au bout de n jours écoulés.

Réponses précédentes : pour tout entier naturel n :

- $v_{n+1} = 0,995v_n + 1,5 \cdot 10^{19}$



- $v_n = 3 \times 10^{21}(0,995^n + 1)$

On souhaite disposer de la liste des termes de la suite $(v_n)_{n \in \mathbb{N}}$.

Pour cela, on utilise une fonction appelée noyaux programmée en langage Python et retranscrite partiellement ci-après.

```
1 def noyaux(n):
2     V = 6 * 10**21
3     L = [V]
4     for k in range(n):
5         V = ...
6         L.append(V)
7     return L
```

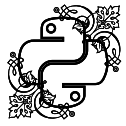
1. À la lecture des questions précédentes, proposer deux solutions différentes pour compléter la ligne 5 de la fonction noyaux afin qu'elle réponde au problème.
2. Pour quelle valeur de l'entier n la commande noyaux(n) renverra-t-elle les relevés quotidiens du nombre de noyaux contenus dans l'échantillon de polonium pendant 52 semaines d'étude?

3.2 Mars 2023 – Amérique du Nord, sujet 2 – Exercice 2.7

• boucle while

- a) Recopier et compléter le programme Python ci-dessous afin qu'il renvoie la plus petite valeur de n à partir de laquelle le nombre de membres du club A est strictement inférieur à 1 280.

```
1 def seuil():
2     n = 0
3     A = 1700
4     while ... :
5         n = n + 1
6         A = ...
```



7 `return ...`

b) Déterminer la valeur renvoyée lorsqu'on appelle la fonction `seui1`.

4. Programme à compléter - impossible à vérifier

Septembre 2023 – Métropole, la Réunion, sujet 2 – Exercice 3.2

- `importer un module`
- `range(a, b)`

On considère la suite (u_n) définie par
$$\begin{cases} u_1 = \frac{1}{e} \\ u_{n+1} = \frac{1}{e} \left(1 + \frac{1}{n} \right) u_n \quad \text{pour tout entier } n \geq 1 \end{cases}$$

On considère une fonction écrite en langage Python qui, pour un entier naturel n donné, affiche le terme u_n . Compléter les lignes 2 et 4 de ce programme

```
1 def suite(n):  
2     ...  
3     for i in range(1, n):  
4         u = ...  
5     return u
```

Remarque : la fonction exponentielle se trouve dans la bibliothèque `math`