

TP12

SIMULATION D'UN PRÊT

Préparer l'exercice 1 – question 1

On souhaite écrire un programme simulant le remboursement d'un prêt à intérêts composés.

Exercice 1 Prenons l'exemple d'un prêt de 10 000 euros avec un taux annuel de 12% et une mensualité de 200 euros.

Le taux annuel de 12% signifie que, chaque mois, un taux d'intérêt de $\frac{12\%}{12} = 1\%$ est appliqué.

- Le premier mois, les intérêts s'élèveront à $10000 \times \frac{1}{100} = 100$ euros.
La mensualité de 200 euros sera donc divisée en :
 - 100 euros d'intérêts
 - 100 euros de remboursement du prêt. Il restera donc 9 900 euros à rembourser. C'est le capital restant dû.
 - Le deuxième mois, les intérêts s'élèveront à $9900 \times \frac{1}{100} = 99$ euros.
La mensualité de 200 euros sera donc divisée en :
 - 99 euros d'intérêts
 - 101 euros de remboursement du prêt. Le capital restant dû est donc de 9 799 euros.
 - et ainsi de suite, jusqu'à ce que le prêt soit remboursé.
1. Compléter le programme suivant pour qu'il affiche, chaque mois, les intérêts payés, le montant remboursé et le capital restant dû, ceci jusqu'au remboursement total.

```
# les données de départ
montant = 10000
taux_annuel = 12/100
mensualite = 200
# ce qui suit doit fonctionner même si l'on change ces données

taux_mensuel = .....

capital_du = .....

mois = 1

while ..... > 0 :

    print('mois ', mois)

    interets = .....

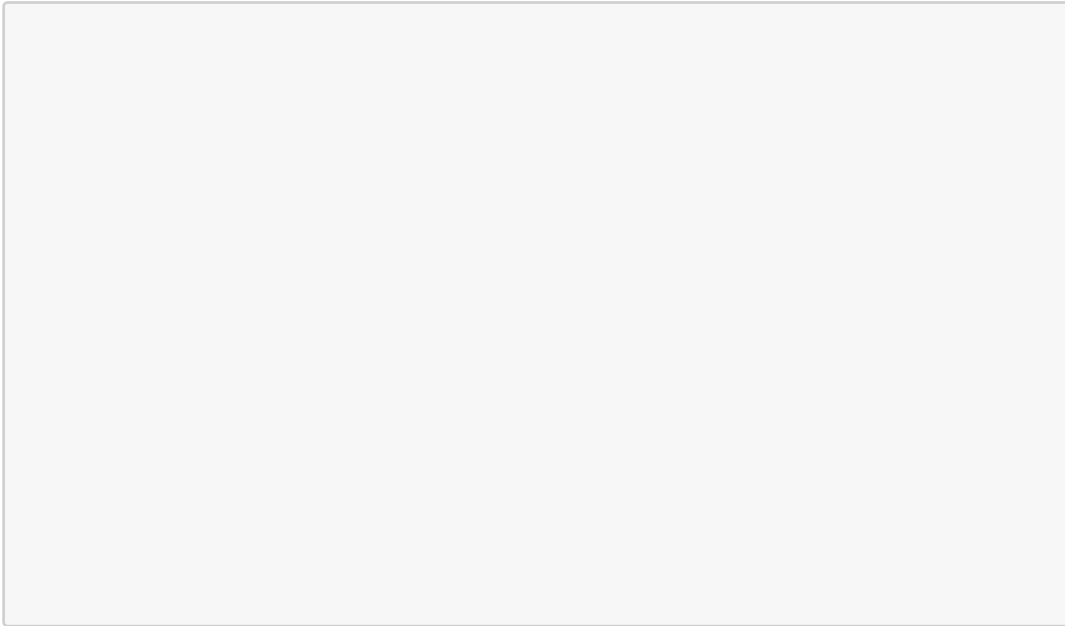
    remboursement = .....

    capital_du = .....

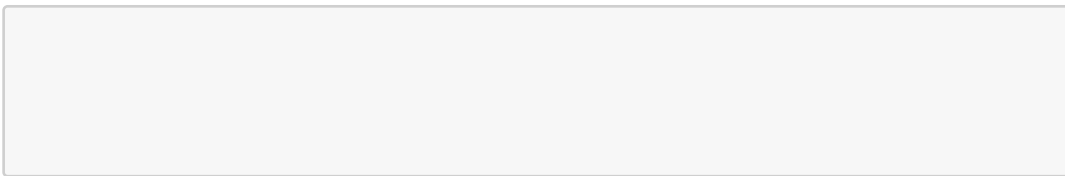
    print('intérêts=', interets, ', remboursement=', remboursement)
    print('capital restant dû =', capital_du)

    mois = .....
```

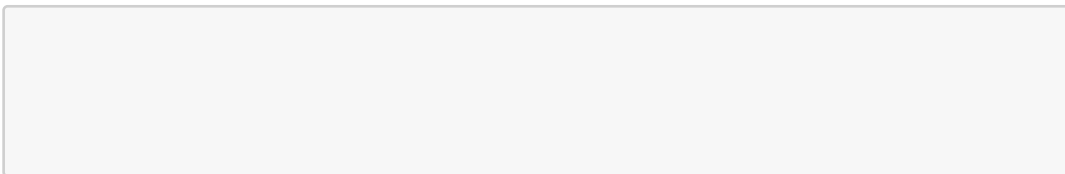
2. Quel est le problème ici? Corriger. Indiquer ici les modifications apportées.



3. Déterminer la durée du prêt, en années et mois.

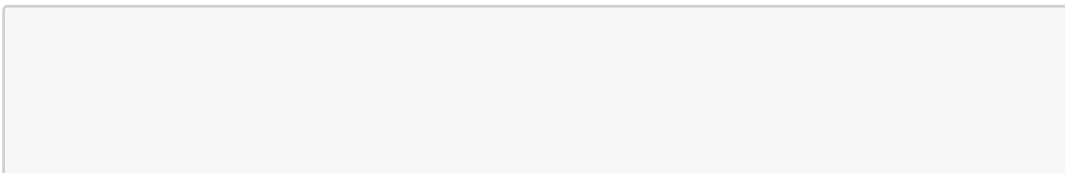


4. Déterminer le coût du prêt, c'est-dire le montant payé à la banque en plus de ce qui vous a été prêté au départ.



5. Écrire une fonction qui, étant donné le montant, la mensualité et le taux annuel, renvoie la durée du prêt (en mois). Inutile de la recopier ici.

6. En général, la banque demande de souscrire une assurance dont le montant est fixe tous les mois. Reprendre le calcul du coût en incorporant une assurance de 10 euros par mois. L'assurance est payée en plus, donc chaque mois l'assuré payera 210 euros.



Exercice 2 Tracer un graphique

Pour tracer des courbes, on va utiliser la bibliothèque `matplotlib.pyplot`, ainsi que `numpy`.

```
import matplotlib.pyplot as plt
import numpy as np
```

Liste des commandes utilisées dans ce TP

Commande	Résultat
<code>plt.plot(x,y)</code>	graphe avec x en abscisse et y en ordonnée
<code>plt.show()</code>	affichage du graphique
<code>np.linspace(a,b,N)</code>	liste de N valeurs allant de a à b

Les abscisses et ordonnées doivent être données sous la forme de listes

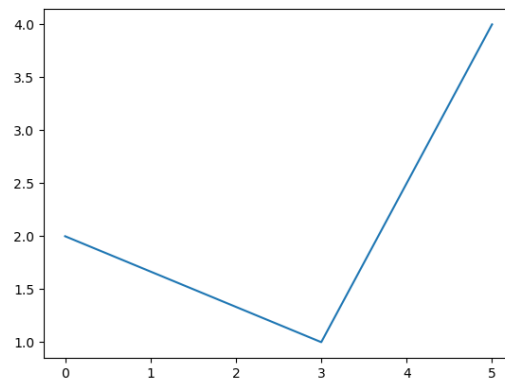
$$x = [x_0, x_1, x_2, \dots, x_n], \quad y = [y_0, y_1, y_2, \dots, y_n].$$

`plt.plot(x,y)` va alors reconstruire les coordonnées et relier les points

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n).$$

Exemple :

```
x = [0, 3, 5]
y = [2, 1, 4]
plt.plot(x,y)
plt.show()
```



On va s'entraîner en tracer la courbe de la fonction \exp sur $[0, 5]$.

1. Pour la liste des abscisses, nous allons avoir besoin de beaucoup de points (100 c'est bien), donc il est plus efficace d'utiliser `np.linspace` que de les écrire à la main.

2. Passons à la liste des ordonnées y . On a $x = [x_0, x_1, x_2, \dots, x_{99}]$ et on veut créer la liste

$$y = [\underbrace{\exp(x_0)}_{y_0}, \underbrace{\exp(x_1)}_{y_1}, \underbrace{\exp(x_2)}_{y_2}, \dots, \underbrace{\exp(x_{99})}_{y_{99}}].$$

Il y a deux méthodes.

- (a) Méthode 1, la plus facile (mais pas toujours possible, on le verra plus tard). On peut directement calculer $\exp(x)$ avec la fonction `exp` de `numpy`.

(b) Méthode 2, avec une boucle for (pour quand la précédente ne fonctionne pas).

```
y = []
for k in range(.....):

    xk = ...

    yk = ...

    y.append(...)
```

3. Enfin, on trace le graphique.

```
plt.plot(x,y)
plt.show()
```

Exercice 3 Durée en fonction du montant

On souhaite tracer le graphique donnant la durée en fonction du montant initial (entre 0 et 100 000 euros).

Ici, les mensualités et le taux annuel sont supposés donnés et fixés.

On ne prend pas en compte l'assurance.

```
taux_annuel = 3/100
mensualite = 500

# Liste des abscisses x (le montant initial)

x =

# Liste des ordonnées y (la durée) avec la méthode 2

# Tracé
plt.plot(x,y)
plt.show()
```

En déduire, par lecture graphique, quel montant on peut emprunter sur 2 ans.