Atelier : Python au quotidien de la classe :

Exemple 1 :

1. Compléter la fonction ci-dessous qui calcule et renvoie le périmètre d'un rectangle de longueur L et de largeur l

```
#fonction perimetre rectangle
def Perimetre_rectangle(L,l):
    perimetre=.......
    return .......
```

2. Ecrire une fonction qui calcule et renvoie l'aire d'un rectangle de longueur L et de largeur l.

Exemple 2 : Un problème de bénéfice.

Une entreprise fabrique et vend des montres. Elle peut en fabriquer jusqu'à 1200 par mois.

Le coût de production mensuel exprimé en euros est pour *n* montres produites :

$$c(n) = 0{,}1n^2 + 100n + 36000$$

L'entreprise vend chaque montre 250 euros pièce.

On admet que la production en un mois est rentable à partir d'un certain nombre de montres produites et vendues.

1. En l'absence de production, l'entreprise a cependant des frais (appelés frais fixes).

A combien s'élèvent-t-ils ?

2. a. Un mois donné, l'entreprise fabrique et vend 200 montres. Est-elle bénéficiaire ?

b. Même question avec 700 montres.

3. On donne ci-dessous le script sous Python de la fonction appelée « cout » qui renvoie le coût de production mensuel pour *n* montres vendues :

```
def cout(n):
    cout=0.1*n*n+100*n+36000
    return cout
```

a. Ecrire de même une fonction que l'on appellera « recette » qui renvoie la recette mensuelle pour *n* montres produites.

b. On donne le script suivant :

```
n=0
while cout(n)> recette(n):
    n=n+1
```

Quelle information permet-il obtenir ?

4. On a recopié sur un fichier Python les fonctions « cout » et « recette » précédentes.

Compléter la fonction **benefice(n)** et le programme ci-dessous afin d'obtenir le nombre de montres que l'entreprise doit produire et vendre pour que son bénéfice mensuel soit maximal.

```
def benefice(n):
    benefice=.......
    return benefice
```

```
bmax=benefice(0)
nmax=0
for k in range(1,1201):
    if benefice(k)>bmax:
        bmax=........
        nmax=........
```

Exemple 3 : L'échiquier

Suivre le lien ... https://lc.cx/YeBY

Exemple 4 : Résoudre un problème

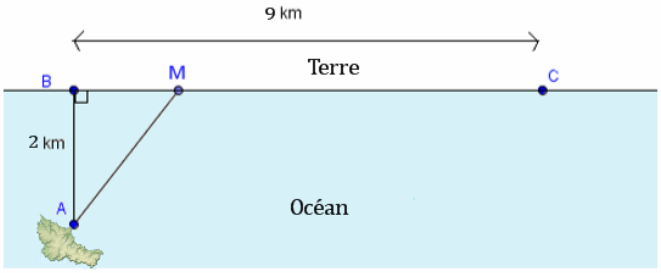
Préambule :

Créer la fonction ***distance*** dont les arguments sont ***v*** et ***t*** et qui renvoie la distance (en km) parcourue lorsqu'on roule à la vitesse ***v*** (en km/h) pendant un temps ***t*** (en h)

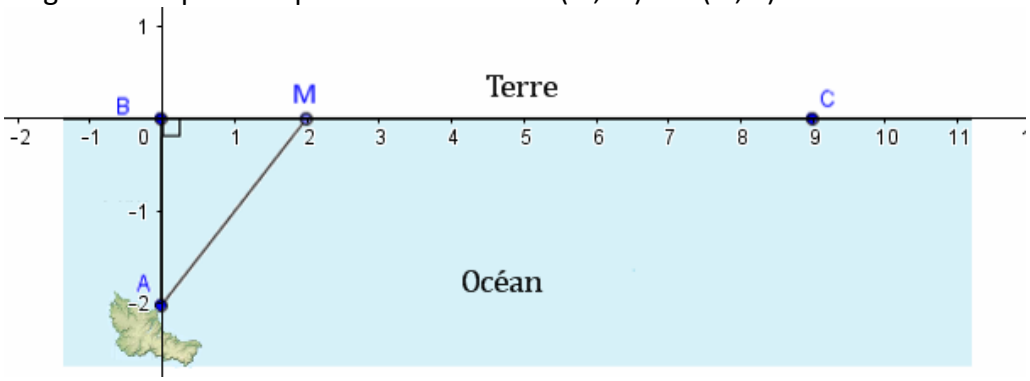
Créer la fonction ***vitesse*** dont les arguments sont ***d*** et ***t*** et qui renvoie la vitesse moyenne lorsqu'on parcourt ***v*** pendant un temps ***t***.

Créer la fonction ***temps*** dont les arguments sont ***v*** et ***d*** et qui renvoie le temps mis pour parcourir une distance ***d*** à la vitesse ***v***.

Où accoster ?



David effectue régulièrement le trajet entre une petite ile, désignée sur le schéma par le point A et la plage des Corsaires, désignée par la lettre C. Sur l'océan, il utilise un canot à moteur.
Il peut accoster n'importe où entre le point B et le point C.
Il termine alors à pied le long de la côte.
On note M le point où il accoste.
Afin d'étudier plus simplement le problème, on représente la situation en se plaçant dans un repère orthonormé d'origine B tel que A ait pour coordonnées A(0 ;–2) et C(9 ; 0).



1) Si David décide d'accoster au point B, quelle distance devra-t-il parcourir au total pour aller de l'île (A) à la plage des Corsaires (C) ?
2) Finalement, il décide d'accoster au niveau du point de coordonnées (2 ; 0). Quelle est alors la distance qu'il parcourt ?
3) Plus généralement, s'il accoste au point M de coordonnées M(x ; 0) avec $0 \leq x \leq 9$, quelle est, en fonction de x la distance qu'il parcourt en mer ?
4) David a écrit cette fonction en Python pour calculer la distance qu'il parcourt en mer.
   a. Cette fonction comporte une erreur. Laquelle ? Corriger-là.
   b. En utilisant ce programme (que vous pouvez compléter) compléter le tableau de valeurs ci-dessous :

```
4 def distance_en_mer(x) :
5     dist_en_mer=x+2
6     return dist_en_mer
7
8 def distance_a_terre(x):
9     return 9-x
```

x	0	1	2	3	4	5	6
Distance totale en fonction de x							

5) Le canot a une vitesse de 4 km/h ; David marche à 5km/h. Jeudi, il a accosté à 2km du point B. Combien de temps a-t-il mis de l'île à la plage des corsaires ?
6) **Question défi :** En modifiant le programme précédent et en utilisant la fonction temps vue dans le préambule, déterminer à 1km près à quel endroit David a intérêt à accoster.

Exemple 5 :

Pierre opère le 01/01/2017 un placement dans sa banque en versant sur un compte 200 euros. La banque rémunère ce compte au taux annuel de 3 %.

Pierre se demande à partir de quelle année il disposera d'au moins 350 euros sur ce compte.

Parmi les programmes suivants, un seul permet de répondre à son interrogation. Lequel ?

Programme1

```
S=200
n=2017
while S>=350:
    S=S*1.03
    n=n+1
```

Programme2

```
S=200
n=2017
while S<350:
    S=S*1.03
```

Programme3

```
S=200
n=2017
while S<350:
    S=S*1.03
    n=n+1
```

Justifier la réponse.

Exemple 6 :

Bref contexte historique

A la cour de Florence au XVIIème siècle, le Duc de Toscane, grand amateur de jeux de dés, avait remarqué à force de jouer, qu'en lançant trois dés et en additionnant la somme des numéros sortis, il obtenait plus souvent une somme égale à 10 qu'une somme égale à 9.

Le Duc ne comprenait pas cette situation car il y a autant de façons d'obtenir la somme égale à 9 que celle égale à 10 :

9 =1+2+6=1+3+5=1+4+4=2+2+5=2+3+4=3+3+3 (6 façons)

10=1+3+6=1+4+5=2+2+6=2+3+5=2+4+4=3+3+4 (6 façons)

Il exposa alors ce paradoxe à Galilée. Ce dernier rédigea en 1620 un mémoire sur les jeux de dés pour répondre au problème du Duc.

1. a. Lancer l'applicationPython. Recopier la fonction ci-contre dans la fenêtre « édition » et tester la

dans la fenêtre « console ».

```
from random import *
def Somme_des():
    de1 = randint(1,6)
    de2 = randint(1,6)
    de3 = randint(1,6)
    somme = de1+de2+de3
    return somme
```

b. Que fait cette fonction ?

c. La réécrire en utilisant une boucle « pour ».

d. Tester à 20 reprises cette fonction et calculer les fréquences d'apparition du 9 et du 10 sur ces 20 lancers.

⟶ mise en commun des résultats obtenus par les élèves ; débat

2. On veut maintenant simuler 1000 parties en utilisant la fonction précédente .

a. Compléter le script suivant qui, pour 1000 lancers de trois dés, comptabilise le nombre d'apparition de la somme 9 et de la somme 10, puis leur fréquence d'apparition sur l'ensemble des 1000 lancers.

```
n=1000
nbr_neuf = 0
nbr_dix = 0
for k in range(n):
    if Somme_des() ==9 :
        nbr_neuf=........
    if Somme_des() == ... :
        ..................

frq_neuf=.............#donne la frequence d'apparition du neuf
frq_dix=.............. #donne la frequence d'apparition du dix
```

b. Ouvrir un nouveau fichier Python . Copier-coller la fonction du **1.** et recopier à la suite le script précédent(complété).

c. Tester à plusieurs reprises dans la fenêtre « console » le programme ainsi créé. Confirmez-vous les observations du Duc de Toscane ?

d. Comment pouvez-vous expliquer le paradoxe du Duc ?

Question défi : Calculer la probabilité d'obtenir le 9 et celle d'obtenir le 10.

Exemple 7 : Fonctions affines

Fonctions affines (signes, variations), tests (if, elif, else), tests imbriqués, utilisation de mathplotlib

Le but du TP est d'étudier les variations et le signe des fonctions de la forme $f(x) = a * x + b$, où a et b sont des réels.

1) Programmation
   a) On a écrit une fonction en python qui renvoie les solutions de l'équation $f(x) = 0$

```
def zeroFonctionAffine(a, b):
    if(a != 0):
        x0 = -b/a
        return x0
    else:
        if(b==0):
            return "R"  #tous les réels
        else:
            return "{}" #ensemble vide
```

   b) Voici une seconde fonction en python que tu dois compléter afin qu'elle renvoie -1 si la fonction est décroissante, +1 si elle est croissante, et 0 sinon

```
def variationFonctionAffine(a):
    if(…):
        return +1
    if(…):
        return -1
    if(….):
        return 0
```

2) Exemple d'application : utilise tes fonctions pour tracer les tableaux de signe des fonctions suivantes :
   a) $u(x) = 3x - 4; v(x) = 15x - 30; p(x) = 1200x + 300$
   b) $f(x) = (2x - 3)(-x + 4); g(x) = (-2x + 5)(-5x + 8)(4x + 2); h(x) = \frac{4x-3}{-2x+1}$
3) **Question défi** :
   Crée une troisième fonction en python, de paramètres a et b, qui doit renvoyer le signe de la fonction affine $f(x) = ax + b$ sous la forme :
   Si a>0 : " - 0 + "
   Si a<0 : " + 0 - "
   Sinon, si b>0 " + + + ", et "- - -" si b<0

Plus loin : Utilisation de matplotlib

En commençant le programme par l'import des bibliothèques :

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

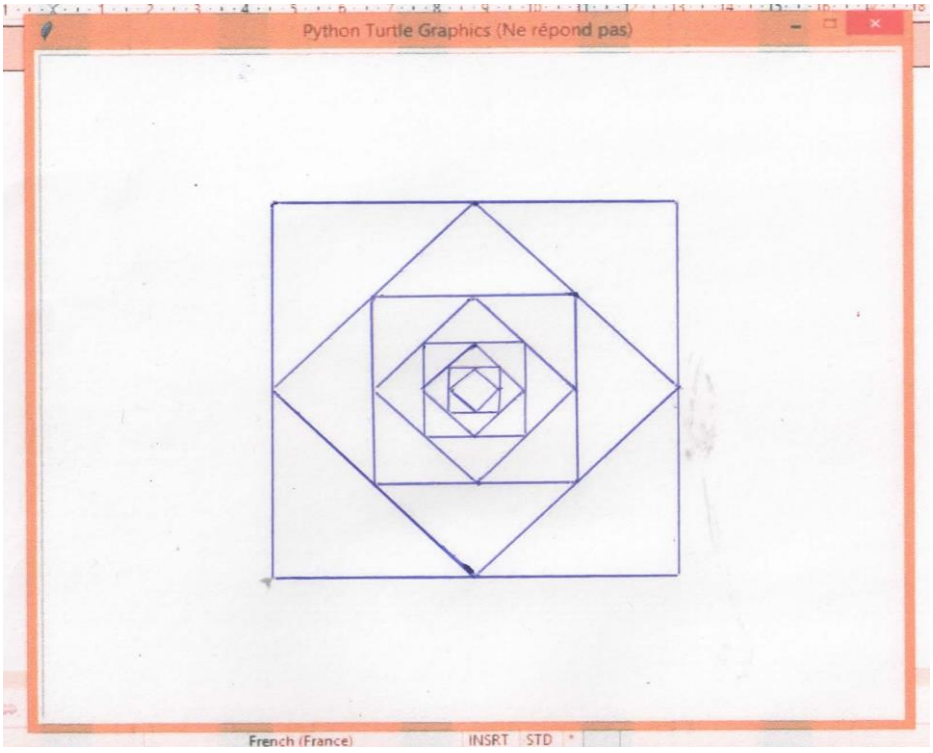
on peut compléter la fonction, ou faire compléter la fonction par l'élève de manière à afficher la fonction autour de sa racine :

```
#A ecrire à la suite de la fonction fonctionAffine :

    xmin=x0-10                          # déterminantion de l'intervalle
    xmax= x0+10                         # xmin-xmax
    x = range(xmin, xmax, 5)            # tableau de 5 valeurs x
    y = a*x+b                           # tableau de 5 valeurs y=ax+b
    plt.plot(x, y)                      # tracer de la droite
    ymin=a*xmin+b                       # calcul de ymin et ymax
    ymax=a*xmax+b                       #
    plt.axvline(0, ymin, ymax,color='k') #axe vertical repère
    plt.axhline(0, xmin, xmax,color='k') #axe horizontal repère
    plt.xlim(xmin,xmax)                 #limitations repère
    plt.ylabel('fonction affine')       #labels
    plt.xlabel("l'axe des abcisses")
    plt.show()                          #affichage du résultat
```

Exemple 8 :

Refaire cette figure (carrés emboités) avec le module Turtle du logiciel Python



Exemple 9 :

Pour résoudre un problème, Sarah a écrit le programme suivant en Python :

Proposer un énoncé de problème possible.

```
def volume(R):
    return 4/3*3.14*R*R*R

R=1

while volume(R)<=20 :
    R=R+0.1
```

Exemple 10 :

Dans une région où il y a autant de femmes que d'hommes, les entreprises sont tenues de respecter la parité. Une entreprise, l'entreprise A, a un effectif de 100 personnes dont 43 femmes.

On admet que respecter la parité revient à ne pas tenir compte du caractère hommes-femme et on considère l'ensemble des salariés d'une entreprise comme un échantillon prélevé au hasard, dans la population de la région

1) Déterminer l'intervalle de fluctuation correspondant à l'entreprise A. Cette entreprise respecte-t-elle la parité ?
2) Une autre entreprise, l'entreprise B, emploie 2500 salariés dont 1150 femmes. Respecte-t-elle la parité ?
3) On dispose désormais des effectifs de plusieurs entreprises de la région.

Entreprise	C	D	E	F	G
Nb de femmes	534	947	301	134	746
Nb total de salariés	1042	1924	674	312	1515

Afin de déterminer si ces entreprises respectent la parité et pour éviter de fastidieux calculs, on souhaite utiliser un algorithme programmé sous python.

a) Compléter la fonction suivante :

```
"""Intervalle de fluctuation"""
from lycee import *


def fluctuation(nb, nbtot, p) :
    freq =
    nbmin=      #nbmin désigne la borne inférieure de l'intervalle de fluctutation
    nbmax=      #nbmax désigne la borne supérieure de l'intervalle de fluctutation
    if(freq>nbmin and freq<nbmax) :
        return True
    else :
        return False
```

b) En utilisant la fonction réalisée, vérifier les résultats des questions 1) et 2).
c) En utilisant la fonction réalisée, déterminer les entreprises qui respectent la parité.

4) **Question défi :** L'entreprise B décide d'embaucher des femmes (sans renvoyer aucun homme) afin de respecter la parité. Combien de femmes doit-elle embaucher au minimum ?

Exemple 11 :

Ecrire une fonction qui calcule le volume d'un cylindre : hauteur h et de rayon de base R.

Exemple 12 :

Ecrire une fonction qui simule le lancer simultané de deux dés parfaitement équilibrés.

Exemple 13 :

Ecrire une fonction qui teste si deux vecteurs dont on connait les coordonnées sont colinéaires.

Exemple 14 :

Créer la fonction ***distance*** dont les arguments sont ***v*** et ***t*** et qui renvoie la distance (en km) parcourue lorsqu'on roule à la vitesse ***v*** (en km/h) pendant un temps ***t*** (en h)

Exemple 15 :

Dans un magasin de reprographie, il existe deux types de photocopieurs. L'affichage du prix total du photocopieur de type A, en fonction du nombre de photocopies réalisées, est géré par une fonction AFFICHAGE codée en Python :

```
def affichage(photocopies):
    if photocopies<=50 :
        prix=photocopies*0.1
    if 50<photocopies and photocopies<=200 :
        prix=5 + (photocopies-50)*0.05
    if photocopies>200 :
        prix=12.5 + photocopies*0.02
    return prix
```

1) Quel est le prix total affiché par le photocopieur A si on réalise 50 photocopies ? 55 photocopies ? 200 photocopies ? 202 photocopies ? (On pourra utiliser la console)
   Cela vous parait-il cohérent ?
2) Rectifier la fonction affichage afin qu'elle indique à l'utilisateur le prix correct.
3) Le photocopieur de type B est réglé ainsi : Il fonctionne à l'aide d'une carte vendue 15€. Cette carte permet d'effectuer 200 photocopies puis à partir de la 201eme, la photocopie est ensuite facturée 0.01 €.
   Un client doit réaliser 400 photocopies. Quel type de photocopieur est-il plus économe d'utiliser ? Même question avec 500 photocopies.
4) Déterminer selon le nombre de photocopies réalisées le type de photocopieur à utiliser.

Question défi

5) Un client a utilisé d'abord le photocopieur A, puis s'est rendu compte qu'il était préférable d'utiliser le photocopieur B.
   Il a dépensé au total 45€.
   Indiquer le nombre possible de photocopies réalisées avec l'appareil A et l'appareil B.

Pour aller plus loin avec Python :

Exemple 16 : Evaluation d'une aire à l'aide de la méthode de Monte Carlo :

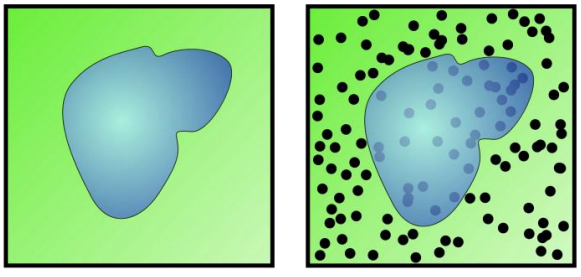
On connait l'aire du rectangle et l'on souhaite estimer la superficie de la zone bleue.

On tire aléatoirement un très grand nombre de points N dans le rectangle. On compte le nombre de points n qui appartiennent à la zone bleue.

Pour un très grand nombre de lancers, on a

$\frac{n}{N} \cong \frac{aire\ zone\ bleue}{aire\ du\ rectangle}$.

Ainsi $aire\ zone\ bleue \cong \frac{n}{N} \times aire\ rectangle$



Exercice :

A l'aide de la méthode de Monte Carlo, estimer l'intégrale

$$J = \int_0^5 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

Quelques fonctions pour l'aléatoire :

```
from math import *
import random
random.random()
```
permet d'obtenir un nombre aléatoire dans l'intervalle [0,1[

Pour visualiser le nuage de point :

```
import matplotlib.pyplot as plt
plt.scatter(x,y,c='red',s=10)
plt.show()
```
permet de placer un point rouge de coordonnées (x,y)

affiche le nuage de points

Exemple 17 :

La suite de Syracuse d'un entier N>0 donné est définie par la relation de récurrence :

$$u_0 = N \text{ et, } \forall\, n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} \text{ si } u_n \text{est pair} \\ 3u_n + 1 \text{ si } u_n \text{est impair} \end{cases}.$$

La conjecture de Syracuse, affirme que pour tout $N \in \mathbb{N}$, il existe un indice *n* tel que $u_n = 1$.

1) Définir une fonction **u(N,n)** qui calcule le n^{ième} terme de la suite de Syracuse qui part de N>0

Vérifiez que pour N=10, on obtient : $u_0 = 10 \quad u_1 = 5 \quad u_2 = 16 \quad u_3 = 8 \quad u_4 = 4 \quad u_5 = 2 \quad u_6 = 1$

2) Représentation graphique : Pour représenter graphiquement la suite u(10,n), pour n variant de 0 à 6, on utilise les instructions:

```
import matplotlib.pyplot as plt
plt.plot([u(10,n) for n in range(6)])
plt.show()
```

Représenter graphiquement la suite Syracuse127 pour un indice n variant de 0 à 50

L'observation graphique de la suite pour N = 127 montre que la suite peut s'élever assez haut avant de retomber. Cela fait penser à la trajectoire d'une feuille emportée par le vent. De cette observation est né tout un vocabulaire imagé : on parlera du vol de la suite. On définit alors :

- Le temps de vol : c'est le plus petit indice n tel que $u_n = 1$. Il est de 6 pour la suite de Syracuse10. On a vu graphiquement que le temps de vol de Syracuse127 est proche de 50.
- L'altitude maximale : c'est la valeur maximale de la suite. Elle est de 16 pour la suite de Syracuse10. Graphiquement, on a vu que Syracuse127 avait une altitude maximale proche de 4500.

3) Définir une fonction **temps_vol(N)** qui a pour argument un entier N>0 et qui renvoie le temps de vol de la suite SyracuseN. Vérifier que temps_vol(10) renvoie 6.

Quelle est la valeur exacte du temps de vol de Syracuse127 ?

4) Définir une fonction **alt_max(N)** qui donne l'altitude maximale de SyracuseN. Vérifier que alt_max(10) renvoie 16. Quelle est la valeur exacte de l'altitude maximale de Syracuse127 ?

5) Ecrire un programme informatique qui démontre la conjecture de Syracuse pour N<1000.