

TP : Algorithmes en PYTHON

Thème 1 : Fonctions & Calculs

25 À la naissance d'Alban, sa grand-mère dépose sur un compte bancaire 100 € et décide d'y verser 100 € à chaque anniversaire auxquels elle ajoute le double de l'âge d'Alban, en euro.

1. Quelle somme totale sera disponible au premier anniversaire d'Alban ? Aux deux ans d'Alban ?
2. Alban a écrit un programme sur sa calculatrice, mais celui-ci comporte une erreur. La retrouver et la corriger.

Casio

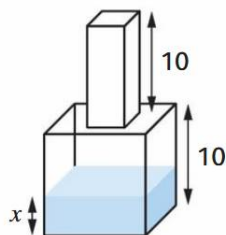
```
100→S
?→A
For 1→I To A
100+2×I→S
Next
S
```

TI

```
:100→S
:Prompt A
:For(I,1,A)
:100+2*I→S
:End
:Disp S
```

3. a. En utilisant le programme corrigé, déterminer à partir de quel âge Alban pourra acheter la guitare de ses rêves d'une valeur de 1 999 € en utilisant l'argent versé par sa grand-mère.
- b. Comment modifier le programme pour qu'il affiche directement la réponse ?

27 Un récipient est formé d'un cube de 10 cm d'arête et d'un parallélépipède rectangle de base carrée de 5 cm de côté et de 10 cm de hauteur.



On le remplit de liquide. On appelle x la hauteur de liquide dans le récipient, en cm.

1. a. À quel intervalle x appartient-il ?
- b. Exprimer le volume $V(x)$ de liquide contenu dans le récipient en fonction de la hauteur x :

$$V(x) = \begin{cases} \dots & \text{si } 0 \leq x \leq 10 \\ \dots & \text{si } \dots \end{cases}$$

c. Utiliser la fonction V pour calculer le volume maximal que peut contenir ce récipient.

2. a. Écrire en langage Python la fonction V . Elle prendra en argument la variable réelle x et renverra le volume de liquide correspondant.

b. On exécute le programme ci-contre, où V est la fonction définie en **2. a.**

```
h=0
while V(h)<=625:
h=h+0.5
```

Quel est son objectif ?

3. En fin d'exécution, la variable h contient la valeur 6,5.

Quelle interprétation peut-on faire de ce nombre dans le contexte de l'exercice ?

28 On considère la fonction f définie sur $[-10; 10]$ par :

$$f(x) = \begin{cases} x^2 + 1 & \text{si } x < 3 \\ \sqrt{97+x} & \text{si } x \geq 3 \end{cases}$$

1. Construire la fonction f en Python.
2. Afficher un tableau de valeurs de f pour les entiers de $[-10; 10]$.

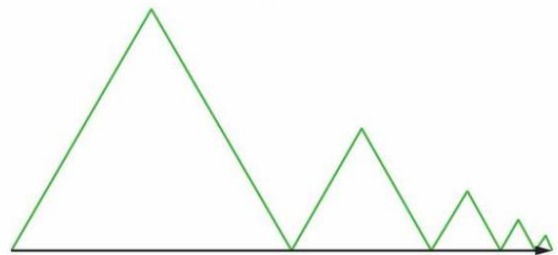
Thème 2 : Géométrie

29 **PYTHON** La fonction `triequi` suivante permet de tracer un triangle équilatéral de côté L pixels.

```
from turtle import*
def triequi(L):
    for i in range(3):
        forward(L)
        left(120)
```

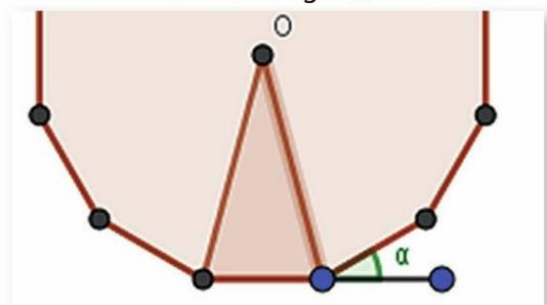
↘ Voir activité 3

La figure ci-après est constituée à gauche d'un triangle équilatéral de côté 200 pixels, puis chaque triangle suivant a pour longueur des côtés la moitié de celle de son voisin de gauche.



1. Construire un programme Python utilisant la fonction `triequi` et permettant de réaliser cette figure.
2. On poursuit la construction en ajoutant, au fur et à mesure à droite, un triangle de moitié de taille. Construire une fonction Python prenant en paramètre le nombre de triangles à construire en tout et réalisant la construction correspondante.

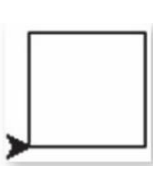
30 1. On considère un polygone régulier à n côtés. Déterminer la valeur de l'angle α en fonction de n .



2. PYTHON Écrire, en Python et à l'aide de sa bibliothèque turtle, la fonction `polyg_regulier` chargée de dessiner un polygone régulier de longueur de côté c et de n côté.

c et n sont des paramètres de la fonction.

Exemples d'exécution :



($c=50$, $n=4$)



($c=20$, $n=10$)

31 Dans un repère du plan, on considère trois points A , B et C , d'abscisses différentes.

1. Dans cette question seulement, on suppose :

$A(1;2)$; $B(5;4)$ et $C(9;6)$

Déterminer par le calcul si les points A , B et C sont alignés.

2. Les points A , B et C ont maintenant pour coordonnées respectives $(x_A; y_A)$, $(x_B; y_B)$ et $(x_C; y_C)$.

En langage Python, cela revient à assimiler chaque point à la liste de ses coordonnées.

Ainsi, x_A sera obtenu par `A[0]` et y_A par `A[1]`.

a. Si on déclare sous Python les variables ci-contre.

```
A=[1,2]
B=[5,4]
C=[9,6]
```

À quoi correspond `B[1]` ?

b. Quelles instructions doit-on saisir pour obtenir l'ordonnée du point A ?

L'abscisse du point C ?

3. On souhaite que la fonction Python `align` ci-contre détermine si les points A , B et C sont alignés ou non et renvoie `True` s'ils sont alignés et `False` sinon.

```
def align(A, B, C):
    m1=(B[1]-A[1])/(B[0]-A[0])
    m2=...
    if ...:
        return True
    else :
        ...
```

a. Interpréter la valeur de la variable `m1` dans le contexte de l'exercice.

b. Compléter la fonction `align`.

4. PYTHON Programmer et tester le programme avec différentes valeurs pour les coordonnées des points A , B et C .

32 Les pyramides mayas peuvent être approximativement considérées comme un empilement de pavés droits à base carrée.



La pyramide de Kukulcán, située au Mexique, comporte neuf étages d'une même hauteur de 2,7 m. L'étage inférieur a une base carrée de côté 55,3 m, puis chaque étage a une base dont le côté est diminué de 2,8 m par rapport au précédent. Cette pyramide est surmontée d'un 10^e étage de 17,2 m de large et de 6 m de haut.

1. Réaliser, à main levée, un schéma de la vue de face de la pyramide. Quelle est sa hauteur totale ?

2. Quel est le volume du 1^{er} étage de la pyramide, c'est-à-dire celui qui repose au sol ? Celui du 2^e étage ? En déduire celui des deux étages.

3. a. **PYTHON** Écrire une fonction en Python qui calcule le volume d'un pavé droit dont les dimensions sont passées en paramètre.

b. À l'aide d'une boucle, déterminer le volume de la pyramide.

Thème 3 : Probabilité Statistiques

33 Obtenir le six, c'est gagné !

On lance un dé cubique équilibré, au plus six fois de suite. Dès que le joueur obtient un 6, il est gagnant et le jeu s'arrête. Si aucun 6 n'apparaît au cours des six lancers, le joueur est perdant.

On souhaite estimer la probabilité de gagner en simulant un grand nombre de parties.

1. On commence par simuler une partie.

a. Expliquer pourquoi il suffit de lancer six fois le dé et de compter le nombre de 6 obtenus sur ces lancers pour savoir si le joueur a gagné.

b. Compléter l'algorithme suivant afin de simuler une partie du jeu.

```
N ← 0
Pour i allant de 1 à 6
    Dé ← Entier_aléatoire(1 ; 6)
    Si ... Alors N ← N + 1
    Fin Si
Fin Pour
Si ... Alors
    Afficher (« Le joueur a gagné. »)
Sinon
    Afficher (« Le joueur a perdu. »)
Fin Si
```

c. Programmer une fonction Python qui simule une partie et qui renvoie `True` si le joueur gagne, et `False` sinon.

Info

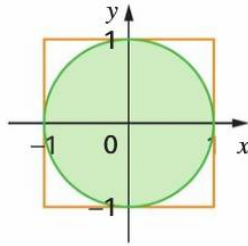
Pour avoir accès aux fonctions liées à la simulation en Python, il faut importer la bibliothèque `random`. L'instruction `randInt(1,6)` permet de générer un entier aléatoire entre 1 et 6.

- Adapter la démarche précédente de façon à simuler n parties, où n est un entier naturel non nul.
- PYTHON** Programmer et estimer la probabilité de gagner à ce jeu.

35 Viser une cible

Sur une planche carrée de côté 2 m, on colorie un disque comme sur la figure ci-contre.

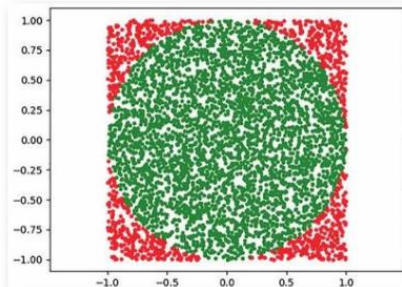
Cette planche devient la cible sur laquelle Medhi envoie des fléchettes au hasard (on suppose qu'il ne rate jamais la planche). On souhaite estimer la probabilité que Medhi atteigne la zone colorée.



- On propose le programme Python ci-dessous.

```
1 from random import*
2 from matplotlib.pyplot import*
3
4 axis([-1,1,-1,1])
5 axis("equal")
6
7 for i in range(5000):
8     x=uniform(-1,1)
9     y=uniform(-1,1)
10    if x**2+y**2<=1:
11        plot(x,y,'.g')
12    else:
13        plot(x,y,'.r')
14
15 show()
16 close()
```

Voici la fenêtre graphique obtenue lorsqu'on exécute ce programme.



Que fait ce programme ? Le mettre en relation avec le contexte de l'exercice.

- En adaptant la démarche précédente, construire une fonction `lancers` qui simule N lancers aléatoires vers la cible et retourne la fréquence d'atteinte de la zone colorée parmi ces N parties, N étant un entier naturel non nul.
- Quelle est la probabilité estimée par Mehdi ?

38 Le jeu du pendu

Pour son projet informatique, Morgane a programmé un « jeu de pendu » en Python. On étudie ici quelques fonctions intervenant dans le code de son programme.

- On souhaite écrire une fonction qui compte le nombre d'occurrences d'une lettre dans un mot donné, c'est-à-dire le nombre de fois où cette lettre apparaît.

```
1 mot_a_deviner="difficile"
2 def occurrence(lettre_proposee,mot):
3     compteur=0
4     for caractere in mot:
5         if caractere==lettre_proposee:
6             compteur= ...
7     return ...
```

Par exemple `occurrence("f",mot_a_deviner)` doit retourner 2. Compléter les éléments manquants de cette fonction, puis la programmer et la tester.

- En utilisant la fonction `occurrence`, écrire une fonction `appartient` qui admet deux arguments : un caractère (`lettre_proposee`) et une chaîne de caractères (`mot`), et qui indique si la lettre proposée appartient ou non au mot.

39 Générateur de mot de passe aléatoire

Pour faciliter la mémorisation de ses mots de passe, Lola souhaite utiliser des syllabes. Sachant qu'il faut souvent utiliser des caractères spéciaux et des chiffres, elle crée trois listes en Python.

```
syllabe = ["mO", "dE", "sU", "tA", "bU", "nA", "tI", "sI"]
caracteres = ['{', '}', '[', ']', '_']
chiffres = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

- En partant d'une chaîne de caractères `mdp` vide tester les instructions ci-contre.

```
1 mdp=""
2 mdp+=syllabe[2]
3 mdp+=chiffres[3]
4 mdp+=caracteres[4]
```

Que contient alors la variable `mdp` ?

À quoi correspond l'instruction `+=` sur une chaîne de caractères ?

- Quelques fois, elle ne souhaite pas intégrer de caractères spéciaux ou de chiffres.

Construire une fonction `motdepasse` qui lui permet de choisir le nombre d'éléments à intégrer dans le mot de passe, de choisir si les caractères spéciaux ou les chiffres sont présents.

Elle souhaite de plus que le chiffre soit placé à la fin de son mot de passe et que le caractère spécial soit au début.

Exemples d'exécution :

```
>>> motdepasse(5,False,False)
'tAbUbUsUdE'
>>> motdepasse(4,True,False)
'}tImOnA'
>>> motdepasse(3,False, True)
'sUnA0'
```