

Algorithme n°1 : l'indice IMC

Ecrire un programme sous python

qui demande la taille (en mètres) et la masse (en kg) d'un individu et lui renvoie sa masse IMC avec un petit commentaire :

Si l'indice IMC est inférieur à 25, le commentaire pourra être : « vous n'êtes pas en surpoids », sinon le commentaire pourra être : « vous êtes en surpoids ».

Cet algorithme utilisera 3 variables : la masse (à transformer en "float"), la taille (à transformer en "float") et $IMC = \frac{masse}{taille \times taille}$

```
from math import *
masse=float(input("Quelle est votre masse (kg) ?"))
taille=float(input("Quelle est votre taille (m) ?"))
IMC=masse/(taille**2)
if IMC<25:
    print("vous n'êtes pas en surpoids")
else:
    print("vous êtes en surpoids")
```

Amélioration n°1 :

Initialiser la masse à -1 et la taille à -1 et intégrer les questions sur la masse et la taille dans une boucle non bornée qui répète la question tant que la réponse est incohérente.

```
from math import *
masse=-1
while not(masse>0 and masse<=635):
    masse=float(input("Quelle est votre masse (kg) ?"))
taille=-1
while not(taille>=0.24 and taille<=2.72):
    taille=float(input("Quelle est votre taille (m) ?"))
IMC=masse/(taille**2)
if IMC<25:
    print("vous n'êtes pas en surpoids")
else:
    print("vous êtes en surpoids")
```

Amélioration n°2

si l'indice IMC est inférieur à 18,5, l'individu est considéré comme « maigre ». Prendre en compte cette information dans les messages.

```
from math import *
masse=-1
while not(masse>0 and masse<=635):
    masse=float(input("Quelle est votre masse (kg) ?"))
taille=-1
while not(taille>=0.24 and taille<=2.72):
    taille=float(input("Quelle est votre taille (m) ?"))
IMC=masse/(taille**2)
if IMC<18.5:
    print("Votre IMC :",IMC," : vous êtes maigre")
else:
    if IMC<25:
        print("Votre IMC :",IMC," : vous n'êtes pas en surpoids")
    else:
        print("Votre IMC :",IMC," : vous êtes en surpoids")
```

Algorithme n°2 : calculer une moyenne

Écrire un programme sous python

qui, à chaque fois qu'on saisit en entrée 5 notes sur 20, affiche en sortie la moyenne de ces cinq notes.

Cet algorithme utilisera au moins les deux variables suivantes :

- la variable note
- la variable moy

```
from lycee import *
total = 0
for i in range(5):
    note=float(input("note : "))
    total=total+note
MOYENNE=total/5
print("la moyenne est",round(MOYENNE,2)," sur 20")
```

Amélioration n°1 :

il est possible de n'utiliser que 2 variables en utilisant au bon endroit l'instruction :

`note=note+float(input("Entrer une note : "))`

```
from lycee import *
note = 0
for i in range(5):
    note=note+float(input("note : "))
print("la moyenne est",round(note/5,2)," sur 20")
```

Amélioration n°2 :

on souhaite maintenant rentrer autant de notes que l'on veut. Comment faire ?

```
from lycee import *
n=int(input("nombre de notes :"))
note = 0
for i in range(n):
    note=note+float(input("note : "))
MOYENNE=note/n
print("la moyenne est",round(MOYENNE,2)," sur 20")
```

Amélioration n°3 :

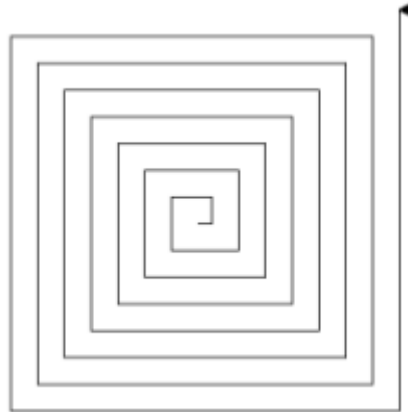
on souhaite que l'utilisateur rentre les notes à partir du message suivant : "Taper une note ou -1 pour terminer"

```
from lycee import *
n, total, note = 0, 0, 0
while note >= 0:
    note = float(input("note ou -1 pour terminer :"))
    if note >= 0:
        total=total+note
        n = n+1
MOYENNE=total/n
print("la moyenne est",round(MOYENNE,2)," sur 20")
```

Algorithme n°3 : la spirale.

Écrire un programme sous python

permettant de créer la spirale ci-dessous.



Elle est composée de 30 segments.

Le premier segment mesure 10 pixels.

A chaque fois que l'on tourne à gauche, le côté du segment augmente de 10 pixels.

Remarque : la première ligne de votre programme intégrera le module « turtle » permettant le dessin :

```
from turtle import *
```

Les instructions utiles :

`forward(longueur)` : pour avancer de la longueur souhaitée en pixels.

`left(90)` : pour tourner à angle droit sur la gauche.

```
from turtle import *
n=10
pendown
for i in range(30):
    forward(n)
    left(90)
    n=n+10
```

Algorithme n°4 : Tirage de dé

Exercice :

L'algorithme ci-dessous permet de lancer un dé à 6 faces 50 fois et d'afficher les résultats :

```
from random import *
for lancer in range (50) :
    print("lancer n°",lancer,":",randint(1,6))
```

Modifier cet algorithme (en gardant le même dé et le même nombre de lancers) pour qu'il affiche le nombre de fois où le 1 est sorti.

```
from random import *
Nombrede1 = 0
for lancer in range (50) :
    Facesortie=randint(1,6)
    if Facesortie == 1 :
        Nombrede1 = Nombrede1 + 1
print ("nombre de 1 :",Nombrede1)
```

Amélioration :

Faire varier le nombre n de tirages et vérifier si la fréquence de sortie du 1 reste dans l'intervalle de fluctuation $\left[p - \frac{1}{\sqrt{(n)}}; p + \frac{1}{\sqrt{(n)}}\right]$

```
from random import *
from math import *
NombreTirages=100
Nombrede1 = 0
for lancer in range (NombreTirages) :
    Facesortie=randint(1,6)
    if Facesortie == 1 :
        Nombrede1 = Nombrede1 + 1
p=1/6
fluct=1/sqrt(NombreTirages)
print("nombre de 1 :",Nombrede1)
freq=Nombrede1/NombreTirages
print("Fréquence : ",freq)
print("intervalle de fluctuation : [",p-fluct,";",p+fluct,"]")
if freq>=p-fluct and freq<=p+fluct:
    ccl="dans"
else:
    ccl="en dehors de"
print("La fréquence est ",ccl," l'intervalle de fluctuation.")
```