

# TestBouclesConditionnelles-Correction

September 23, 2020

## 1 Structure conditionnelle

### 1.1 Exercice 1

```
[1]: masse_utile = 1800
masse = int(input('Masse d\'un poteau ? '))
nombre = int(input('Nombre de poteaux ? '))
if nombre * masse > masse_utile:
    print('Surcharge !')
```

Masse d'un poteau ? 100

Nombre de poteaux ? 19

Surcharge !

### 1.2 Exercice 2

```
[2]: texte = input("Saisir un texte : ")
n = 0 #compteur du caractère 'e'
for c in texte:
    if c == 'e':
        n = n + 1
print("Votre texte comporte ", n, "caractères 'e'")
```

Saisir un texte : ineptie

Votre texte comporte 2 caractères 'e'

### 1.3 Exercice 3

```
[4]: n = int(input('Entrez un nombre entier naturel : '))
S = 0
for k in range(1, n):
    if n % k == 0:
        S = S + k
if S == n:
    print(n, "est parfait")
else:
    print(n, "n'est pas parfait")
```

Entrez un nombre entier naturel : 6  
6 est parfait

## 1.4 Exercice 4

```
[18]: from random import random
      from math import sqrt

      x = random()
      y = random()
      print(x, y)
      if sqrt(x ** 2 + y ** 2) < 1:
          print("Gagné")
      else:
          print("Perdu")
```

0.7964167759799121 0.7177039673069944  
Perdu

## 1.5 Entraînement 1

```
[20]: from random import random
      from math import sqrt

      victoire = 0
      for k in range(4000000):
          x = random()
          y = random()
          if sqrt(x ** 2 + y ** 2) < 1:
              victoire = victoire + 1
      print(victoire / 1000000)
```

3.142183

### Observation :

On obtient une fréquence proche de  $\pi$ .

### Explication :

La probabilité de victoire est égale au rapport de l'aire du disque de rayon 1 à l'aire du carré de côté 2 représentant les points de coordonnées  $(x, y)$  telles que  $-1 \leq x \leq 1$  et  $-1 \leq y \leq 1$ . Cette probabilité est donc de  $\frac{\pi}{4}$ .

La fréquence de victoires sur un échantillon de taille 4000000 s'obtient par le calcul  $\frac{\text{nombrede victoires}}{4000000}$ . D'après la loi faible des grands nombres on a  $\frac{\text{nombrede victoires}}{4000000} \approx \frac{\pi}{4}$ .

On en déduit que  $\frac{\text{nombrede victoires}}{1000000} \approx \pi$ .

## 1.6 Exercice 5

```
[22]: n = int(input("Entrez le nombre de l'année : "))

if n % 4 != 0:
    print("Année non bissextile")
elif n % 100 != 0:
    print("Année bissextile")
elif n % 400 != 0:
    print("Année non bissextile")
else:
    print("Année bissextile")
```

Entrez le nombre de l'année : 2020

Année bissextile

## 1.7 Exercice 7

```
[ ]: (x_A, y_A) = (1, 5)
(x_B, y_B) = (4, 3)
x = float(input("Entrez l'abscisse de M"))
y = float(input("Entrez l'ordonnée de M"))
#dans les conditions on envisage un cas général pour les coordonnées de A et B
if (x_A <= x <= x_B or x_B <= x <= x_A) and (y_A <= y <= y_B or y_B <= y <= y_A):
    print("M est dans le rectangle de sommets opposés A et B")
else:
    print("M n'est pas dans le rectangle de sommets opposés A et B")
```

## 1.8 Exercice 8

```
[ ]: (x_A, y_A) = (2, -3)
r = 4
x = float(input("Entrez l'abscisse de M"))
y = float(input("Entrez l'ordonnée de M"))
#dans les conditions on envisage un cas général pour les coordonnées de A et B
if (x - x_A) ** 2 + (y - y_A) ** 2 <= r ** 2:
    print("M est dans le disque de centre A et de rayon r")
else:
    print("M n'est pas dans le disque de centre A et de rayon r")
```

## 1.9 Entraînement 3

```
[25]: n = int(input('Saisir un entier n : '))
premier = True
for k in range(2, n // 2):
    premier = premier and n % k != 0
```

```

if premier:
    print(n, " est premier")
else:
    print(n, " n'est pas premier")

```

Saisir un entier n : 33  
 33 n'est pas premier

```

[26]: #en pratique si d divise n alors n = n // d * d et donc max(d, n // d) ** 2 <= n
from math import sqrt
n = int(input('Saisir un entier n : '))
premier = True
for k in range(2, int(sqrt(n)) + 1):
    premier = premier and n % k != 0
if premier:
    print(n, " est premier")
else:
    print(n, " n'est pas premier")

```

Saisir un entier n : 49999  
 49999 est premier

### 1.10 Exercice 10

```

[31]: n = int(input('Saisir un entier 1<= n <= 100 :'))
while not(1 <= n <= 100):
    n = int(input('Saisir un entier 1<= n <= 100 :'))
print("Vous avez saisi : ", n)

```

Saisir un entier 1<= n <= 100 :101  
 Saisir un entier 1<= n <= 100 :-2  
 Saisir un entier 1<= n <= 100 :2  
 Vous avez saisi : 2

### 1.11 Entraînement 4

```

[34]: password = input('Saisir le mot de passe : ')
while not(input('Mot de passe ? ') == password):
    print('Mot de passe invalide, nouvel essai')
print("Connexion réussie")

```

Saisir le mot de passe : secret  
 Mot de passe ? abcd  
 Mot de passe invalide, nouvel essai  
 Mot de passe ? secret  
 Connexion réussie

### 1.12 Exercice 11

```
[35]: x = float(input("Saisir un réel x >= 1 "))
while not(x >= 1):
    x = float(input("Saisir un réel x >= 1 "))
n = 1
x = x // 2
while x >= 1:
    n = n + 1
    x = x // 2
print("Le logarithme entier de ", x, " est ", n)
```

Saisir un réel x >= 1 16

Le logarithme entier de 0.0 est 5

### 1.13 Exercice 12

```
[43]: motif = 'NSI'
chaine = input('Saisir une chaîne de caractères : ')
i = 0
trouve = False
while i <= len(chaine) - 3 and not trouve:
    j = 0
    while j < len(motif) and chaine[i + j] == motif[j]:
        j = j + 1
    if j == len(motif):
        trouve = True
    i = i + 1
if trouve:
    print("'NSI' trouvée en position ", i)
else:
    print("'NSI' ne se trouve pas dans la chaîne")
```

Saisir une chaîne de caractères : ABNSI

'NSI' trouvée en position 3