

## Examples :

Au bowling, on a deux chances d'écrire un algorithme qui démarre chaqueune des deux boules et affiche la première boule, " / " si toutes les deux sont égales et renvoie le nombre de quille renversées.

## ACTIVITÉ 2

```

score = int(input("Enter 1st score : "))
gain = int(input("Enter 2nd gain : "))
nouveau_score = score + gain
print("New score = " + str(nouveau_score))

if nouveau_score == 50:
    print("Victoire")
else:
    print("Nouveau score < 50 :")
    print("Nouveau score = " + str(nouveau_score))
    if nouveau_score < 50:
        print("Nouveau score : " + str(nouveau_score))
    else:
        print("Nouveau score : 25")

```

CORRECTION

Au jeu de molkky, chaque joueur marquera à son tour de jeu entre 0 et 12 points, qui viennent s'ajouter à son score précédent. Le premier à atteindre un score de 51 gagne. Mais garé ! Quiconque dépasse le score cible de 51 revient immédiatement à 25 points.

Écrire un algorithme demandant un score "Entrez le score : " et un nombre de points marqués "Entrez le score : ", et qui affiche le nouveau score "Nouveau score : " ou signale une éventuelle victoire "Victoire".

## ACTIVITÉ 1

## 6.4 – Activities

```

    n = 0
    k = int(input("Entrer l'ordre du code : "))
    for _ in range(k):
        t = int(input("Saisir le Lettre : "))
        if t == 1:
            print("A")
        elif t == 2:
            print("B")
        elif t == 3:
            print("C")
        elif t == 4:
            print("D")
        elif t == 5:
            print("E")
        else:
            print("Lettre non reconnue")

```

CORRECTION

Ecrire un programme qui demande à l'utilisateur la valeur de l'ordre k, puis demande de saisir k lettres. Le programme affiche la lettre majoritaire, ou X si aucune n'est majoritaire.

Exemples :

```
>>> Entrer l'ordre du code : 5
>>> Saisir le caractère : 0
>>> Saisir le caractère : 1
>>> Saisir le caractère : 1
>>> Saisir le caractère : 1
>>> Saisir le caractère : 0
>>> Entrer l'ordre du code : 5
>>> Saisir le caractère : 0
>>> Saisir le caractère : 1
>>> Saisir le caractère : 0
```

```
>>> Score première boule : 2
>>> Score deuxième boule : 3
5

>>> Score première boule : 7
>>> Score deuxième boule : 3
/
>>> Score première boule : 10
>>> Score deuxième boule : 0
X
```

**CORRECTION**

```
[ ]: score_1 = int(input("Score première boule : "))
score_2 = int(input("Score deuxième boule : "))
if score_1 == 10:
    print("X")
elif score_1 + score_2 == 10:
    print("/")
else:
    print(score_1 + score_2)
```

 **ACTIVITÉ 3**

Implémenter un programme qui demande deux nombres à l'utilisateur et affiche "divisible" si le premier est divisible par le second et qui affiche "pas divisible" sinon.

**CORRECTION****Exemples :**

```
>>> Première lettre : 0
>>> Deuxième lettre : 1
>>> Troisième lettre : 1
1

>>> Première lettre : 1
>>> Deuxième lettre : 0
>>> Troisième lettre : 0
0
```

**CORRECTION**

```
[ ]: l1 = input("Première lettre : ")
l2 = input("Deuxième lettre : ")
l3 = input("Troisième lettre : ")

if l1 == l2 or l1 == l3:
    print(l1)
else:
    print(l3)
```

 **ACTIVITÉ 11**

Pour limiter les erreurs dans la transmission d'un message, une technique simple appelée code de répétition d'ordre  $k$  consiste à répéter  $k$  fois chaque lettre.

Lors de la réception du message, il suffit alors de regarder quelle lettre est majoritaire dans chaque paquet de lettres.

Dans cet exercice, on ne considérera comme lettre que les nombres 0 et 1.

CORRECTION

<<< Entrer un nombre entier : 10

```
>>> Enterer un nombre entier : 26  
26  
13  
2  
1  
1
```

### Examples :

٤٣٦

Ecrire un programme qui demande un entier "Entrer un nombre entier : " à l'utilisatuer et affiche tous ses diviseurs.

**ANSWER** The answer is 1000.

```
print("pas divisible")
else:
    print("divisible")
    if a % b == 0:
        b = int(input("saisir un premier nombre: "))
    else:
        a = int(input("saisir un premier nombre: "))

[ ]
```

Ecrire un programme qui demande à l'utilisateur de saisir trois lettres et qui affiche la lettre majoritaire.

Dans cet exercice, on ne considéera comme lettre que les nombres 0 et

Lors de la réception du message, il suffit alors de regarder quelle lettre est majoritaire dans chaque paquet de lettres.

Pour limiter les erreurs dans la transmission d'un message, une technique simple appelle code de répétition d'ordre k consiste à répéter k fois chaque lettre.

## ACTIVITÉ 10

```
    [ ] : 11 = Input("Permitte Lettere : ")  
    12 = Input("Deutsche Lettre : ")  
    if 11 == 12:  
        print(11)  
    else:  
        print("X")
```

CORRECTION

>>> Première Lettre : I  
>>> Deuxième Lettre : I

v

```
>>> Première Lettre : O  
>>> Deuxième Lettre : I
```

<<> Deuxième Lettre : 0


**ACTIVITÉ 5**

Écrire un programme qui demande un entier  $n$  à l'utilisateur et affiche tous ses diviseurs en précisant à la fin de l'affichage si le nombre est premier ou pas.

Exemples :

```
>>> Entrer un nombre entier positif : 26
```

```
1
```

```
2
```

```
13
```

```
26
```

```
26 n'est pas premier
```

```
>>> Entrer un nombre entier positif : 41
```

```
1
```

```
41
```

```
41 est premier
```

**CORRECTION**

```
[ ]: n = int(input("Entrer un nombre entier positif : "))
if n < 0:
    print(n, "n'est pas positif")
else:
    nb_diviseurs = 2
    print(1)
    for i in range(2, n//2 + 1):
        if n % i == 0:
            nb_diviseurs = nb_diviseurs + 1
            print(i)
    print(n)
if nb_diviseurs == 2:
    print(n, "est premier")
else:
    print(n, "n'est pas premier")
```

```
[ ]: a = int(input("Entrer la première distance : "))
b = int(input("Entrer la deuxième distance : "))
c = int(input("Entrer la troisième distance : "))
if a+b >= c and b+c >= a and c+a >= b:
    print("Ceci est un triangle")
    if a == b and b == c:
        print("équilatéral")
    elif a == b or b == c or c == a:
        print("isocèle")
    else:
        print("scalène")
    if a*a+b*b == c*c or b*b+c*c == a*a or c*c+a*a == b*b:
        print("rectangle")
else:
    print("Ceci n'est pas un triangle")
```


**ACTIVITÉ 9**

Pour limiter les erreurs dans la transmission d'un message, une technique simple appelée code de répétition d'ordre  $k$  consiste à répéter  $k$  fois chaque lettre.

Lors de la réception du message, il suffit alors de regarder quelle lettre est majoritaire dans chaque paquet de lettres.

Dans cet exercice, on ne considérera comme lettre que les nombres 0 et 1.

Écrire un programme qui demande à l'utilisateur de saisir deux lettres et qui affiche la lettre majoritaire, ou X si aucune n'est majoritaire.

Exemples :

```
>>> Première lettre : 0
```

```
>>> Deuxième lettre : 0
```

```
0
```

```
>>> Première lettre : 1
```

Au bowling, on a deux chances pour faire tomber un total de dix quilles. Ecrire un programme qui demande le nombre de quilles renversées avec la première des deux boules et affiche le nombre de quilles renversées à chaque boule, / si toutes les quilles sont tombées, et sinon le nombre de quille renversées.

Améliorer votre programme en ne demandant les informations de la deuxième boule que si elle a besoin d'être lancée.

Exemples :

<<>> Score première boule : 2

<<>> Score deuxième boule : 3

<<>> Score deuxième boule : 5

<<>> Score première boule : 7

<<>> Score deuxième boule : 3

<<>> Score première boule : 10

/

TE 6

November 2022

2 - PROGRAMMER EN PYTHON

LANGUAGE

```
score_1 = int(input("Score première boule : "))
if score_1 == 10:
    print("Score première boule : ")
    print("X")
else:
    print(score_1)
score_2 = int(input("Score deuxième boule : "))
if score_2 == 10:
    print("Score deuxième boule : ")
    print("X")
else:
    print(score_2)
print(score_1 + score_2)
```

CORRECTION

CORRECTION

## rectangle scalene

Ceci est un triangle

>>> Entrer La troisième distance : 5  
>>> Entrer La deuxième distance : 4  
>>> Entrer La première distance : 3

## équilateral

Ceci est un triangle

>>> Entrer La première distance : 5  
>>> Entrer La deuxième distance : 5  
>>> Entrer La troisième distance : 5

ÍSOCELE

Ceci est un triangle

>>> Entrer La première distance : 3  
>>> Entrer La deuxième distance : 3  
>>> Entrer La troisième distance : 5

## Scaléne

>>> Emüter La troisème distance : 6  
>>> Ceci est un triangle

novembre 2021

2 - PROGRAMMER EN PYTHON

---

LANGUAGE


**ACTIVITÉ 7**

Au bowling, on a deux chances pour faire tomber un total de dix quilles. Écrire un programme qui demande le nombre de quilles renversées avec chacune des deux boules et affiche X si toutes les quilles sont tombées à la première boule, / si toutes les quilles sont tombées, et sinon le nombre de quille renversées.

Améliorer votre programme en ne demandant les informations de la deuxième boule que si elle a besoin d'être lancée.

Améliorer votre programme en affichant ! si les scores saisis sont impossibles.

Exemples :

```
>>> Score première boule : 2
>>> Score deuxième boule : 3
5
```

```
>>> Score première boule : 7
>>> Score deuxième boule : 3
/
```

```
>>> Score première boule : 10
X
```

```
>>> Score première boule : -5
!
```

```
>>> Score première boule : 13
!
```

```
>>> Score première boule : 2
>>> Score deuxième boule : -3
!
```

```
>>> Score première boule : 2
>>> Score deuxième boule : 9
!
```

**CORRECTION**

```
[ ]: score_1 = int(input("Score première boule : "))
if score_1 < 0 or score_1 > 10:
    print("!")
elif score_1 == 10:
    print("X")
else:
    score_2 = int(input("Score deuxième boule : "))
    if score_2 < 0 or score_1 + score_2 > 10:
        print("!")
    elif score_1 + score_2 == 10:
        print("/")
    else:
        print(score_1 + score_2)
```


**ACTIVITÉ 8**

Écrire un programme qui demande trois longueurs à l'utilisateur, indique si ces trois longueurs peuvent être les longueurs des trois côtés d'un triangle et, le cas échéant, s'il s'agit d'un triangle équilatéral, isocèle, rectangle ou scalène (trois côtés de longueurs différentes)

Exemples :

```
>>> Entrer la première distance : 2
>>> Entrer la deuxième distance : 3
>>> Entrer la troisième distance : 10
Ceci n'est pas un triangle
```