NSI Interrogation n°1 26/09

Exercice 1

Indiquer la valeur finale de la variable b dans les algorithmes suivants :

Algorithme 1	Algorithme 2

`a=3`
`b=6`
`if a>5 or b!=3:`
`    b=4`
`else:`
`    b=2`
	`a=7`
	`b=12`
	`if a>5:`
	`    b=b-4`
	`if b>=10:`
	`    b=b+1`

Valeur de *b* :	**Valeur de *b* :**
Algorithme 3	**Algorithme 4**

`a=2`
`b=5`
`if a>8:`
`    b=10`
`elif a>6:`
`    b=3`
	`a=2`
	`b=0`
	`if a<0:`
	`    b=1`
	`elif a>0 and a<5:`
	`    b=2`
	`else:`
	`    b=3`

Valeur de *b* :	**Valeur de *b* :**

Exercice 2

Indiquer pour chaque fragment de code le résultat affiché en sortie

Code	Résultat en sortie ?

`s = 0`
`for i in range(1,4):`
`    s = s + 1`
`print(s)`	

`n = 0`
`while n<15 :`
`    n = n + 2`
`print(n)`	

`a= 26`
`b= 3`
`q = 0`
`while a > b:`
`    q = q + 1`
`    a = a - b`
`print(q , a)`	

`res = 1`
`for i in range(3):`
`    res = res * 2`
`    print(res)`	

`n = 10`
`while n>=11 :`
`    n = n + 2`
`print(n)`	

Exercice 3

1. Compléter le programme ci-contre pour qu'il affiche le montant présent sur un compte d'épargne rémunéré avec un taux de 3.5% pendant 15 ans, sachant que la somme initialement versée sur le compte est 1500€.

```
s = 1500
for i in range(15):
    s = _____________________
print(s)
```

2. Modifier ce programme pour qu'il affiche cette fois le nombre d'années nécessaires pour atteindre au moins 5000 euros sur le compte d'épargne

Programme modifié :

Exercice 4

Ecrire ci-dessous une fonction *factorielle(n)* qui prend en paramètre un entier naturel, et renvoie en sortie la valeur 1 si $n = 0$ ou la valeur $1 \times 2 \times 3 \times \ldots \times n$ sinon

Exercice 5

Dans chacun des cas suivant, donnez la valeur de la variable booléenne *rep* :

Définition de la variable booléenne *rep*	**Valeur de *rep***	**Définition de la variable booléenne *rep***	**Valeur de *rep***
`x = -3` `rep = xx*2==-9`		`a, b = 3, -7` `rep= a**3 > 50 and b**2 < 50`	

`x, y, z = 3, 4, 5`
`rep = x**2 + y**2 == z**2`
		`a, b = 3, -7`
		`rep= (a**3 > 50 and b**2 < 50) or`
		`(a**2 < 10 and b**2 > 10)`	

`x, y, z = 5, 5, 10`
`rep = x**2 + y**2 == z**2`
		`from math import sqrt`
		`x, y, z = 3, 3, sqrt(18)`
		`rep = x**2 + y**2 == z**2`	

Exercice 6

1. On veut écrire une fonction *triangle(c)* qui trace un triangle équilatéral de côté *c*. Indiquer **laquelle** des propositions ci-dessous est correcte ***(cocher la bonne case)***

Proposition A :

```
def triangle(c):
    c = int(input("entrez une longueur :"))
    for i in range(3):
        forward(c)
        left(60)
```

Proposition C :

```
def triangle(c):
    for i in range(1,3):
        forward(c)
        left(60)
```

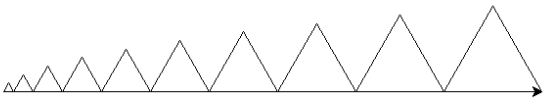
Proposition B :

```
def triangle(c):
    for i in range(3):
        forward(c)
        left(120)
```

Proposition D :

```
def triangle(c):
    for i in range(3):
        forward(c)
        left(60)
```

2. En utilisant la fonction *triangle(c)*, écrire un programme qui reproduit le dessin ci-contre : 10 triangles équilatéraux, premier triangle de côté 10 pixels, le côté augmente de 10 pixels à chaque fois



Programme :

Exercice 6

On souhaite écrire une fonction *constructible(a,b,c)* qui prend en paramètre les longueurs de trois côtés et renvoie *True s*'il est possible de construire le triangle correspondant, *False* sinon.

Parmi les propositions suivantes, indiquer TOUTES CELLES qui conviennent ***(cocher les bonnes cases)***

```
def constructible(a,b,c):
    if a+b>c or a+c>b or b+c>a:
        return True
    else:
        return False
```

```
def constructible(a,b,c):
    return (a+b<c and a+c<b and b+c<a)
```

```
def constructible(a,b,c):
    return (a+b>c and a+c>b and b+c>a)
```

```
def constructible(a,b,c):
    return (a+b>c or a+c>b or b+c>a)
```

```
def constructible(a,b,c):
    return (a+b<c or a+c<b or b+c<a)
```

```
def constructible(a,b,c):
    if a+b<c or a+c<b or b+c<a:
        return False
    else:
        return True
```

```
def constructible(a,b,c):
    if a+b<c and a+c<b and b+c<a:
        return False
    else:
        return True
```

```
def constructible(a,b,c):
    if a+b>c and a+c>b and b+c>a:
        return True
    else:
        return False
```

Exercice 7

On veut programmer un jeu qui s'apparente au BlackJack mais se joue avec deux dés. Le joueur est opposé à la banque, **le but est de s'approcher de 21 sans dépasser ce total.**

A chaque tour, le joueur choisit de lancer 0, 1 ou 2 dés.

- *S'il choisit 0, la partie s'arrête*
- *Sinon, on calcule le score total aux dés. Le banquier lance alors le même nombre de dés que le joueur.*
  - *Si l'un des deux dépasse 21, il a perdu et l'autre a gagné*
  - *Si les deux dépassent 21, la partie est déclarée nulle*
  - *Sinon, le joueur peut continuer à lancer à nouveau un certain nombre de dés pour se rapprocher de 21*

Compléter le script ci-dessous pour le programme réponde au cahier des charges qui simule ce jeu.

```
from random import *
TotalJoueur=0
TotalBanque = 0
continuer = True
while continuer==True :
    n=int(input("Combien de dés ?"))
    if n > 0 :
        d = 1
        print ("au joueur de lancer !")
        while  d <= n :
            res=randint(..............)
            print ("Dé numéro", d, ":", res)
            TotalJoueur = ...................
            d = d + 1
        print ("Total du joueur", TotalJoueur)
        print ("Au tour de la banque de lancer :")
        d = 1
        while d <= n :
            res = randint(..............)
            print ("Dé numéro", d, ":", res)
            ..................................
            d = d + 1
        print ("Total de la banque", TotalBanque)
        if TotalBanque > 21 ........ TotalJoueur > 21 :
            continuer = False
    else :
        continuer = False
# On regarde les résultats :

if TotalJoueur > 21 :
    if TotalBanque .............. :
        print ("Match nul")
    else :
        print ("Vous avez perdu, vous avez dépassé 21.")
else :
    if TotalBanque > 21 :
        print ("Vous avez gagné, la banque a dépassé 21.")
    ......................
    ......................
    ......................
    ......................
    ......................
    ......................
```