
Préparation à l'évaluation 1^{ère} NSI
Correction

Question 1

Utilisation de range() dans une boucle for

La fonction range() est souvent utilisée pour contrôler le nombre d'itérations dans une boucle for.

Exemple 1 : Utilisation de range(3)

```
for i in range(3):  
    print(i)
```

Exemple 2 : Utilisation de range(3, 7)

```
for i in range(3, 7):  
    print(i)
```

Exemple 3 : Utilisation de range(1, 8, 2)

```
for i in range(1, 8, 2):  
    print(i)
```

Exemple 4 : Utilisation de range(10, -1, -2)

```
for i in range(10, -1, -2):  
    print(i)
```

Pour chaque exemple, écrivez les valeurs de i qui seront affichées à l'écran.

```
for i in range(3):  
    print(i)
```

0

1

2

```
for i in range(3, 7):  
    print(i)
```

3

4

5

6

```
for i in range(1, 8, 2):
```

```
    print(i)
```

1

3

5

7

```
for i in range(10, -1, -2):
```

```
    print(i)
```

10

8

6

4

2

0

Question 2

On définit $L = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15]]$. Quelle est la valeur de $L[0][2]$?

Réponses

A 2

B 3

C 11

D 12

Réponse : B

Question 3

On définit : `matrice = [[1,2,3], [4,5,6], [7,8,9], [10,11,12]]`.
Quelle est la valeur de `matrice[1][2]` ?

Réponses

- A 2
- B 4
- C 6
- D 8

Réponse : C

Question 4

On considère le programme suivant :

```
Somme = 0
for i in range(4):
    somme = somme + i
print(somme)
```

Dérouler l'algorithme dans le tableau ci dessous

somme							
i							

Correction :

```
i= 0
somme= 0
-----
i= 1
somme= 1
-----
i= 2
somme= 3
-----
i= 3
somme= 6
-----
```

Question 5

L est une liste d'entiers.

On définit la fonction suivante :

```
def f(L):  
    m = L[0]  
    for x in L:  
        if x > m:  
            m = x  
    return m
```

Que calcule cette fonction ?

Réponses

- A le maximum de la liste L passée en argument
- B le minimum de la liste L passée en argument
- C le premier terme de la liste L passée en argument
- D le dernier terme de la liste L passée en argument

Réponse : A le maximum de la liste L

Question 6

On définit :

```
L = [ ["lundi",10,0.87], ["mardi",11,0.82], ["mercredi",12,0.91] ]
```

Quel est le type de la variable a définie par a = L[1][2] ?

Réponses

- A nombre entier
- B liste
- C nombre flottant
- D chaîne de caractères

Réponse : C Nombre flottant

Question 7

Quel est le type de l'expression f(4) si la fonction f est définie par :

```
def f(x):  
    return (x, x**2)
```

Réponses

- A un entier
- B un flottant
- C une liste
- D un tuple

Réponse : Un tuple

Question 8

On exécute le code suivant :

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
for i in range(3):
    for j in range(3):
        B[i][j] = A[j][i]
```

Que vaut B à la fin de l'exécution ?

Réponses

- A rien du tout, le programme déclenche une erreur d'exécution
- B [[3, 2, 1], [6, 5, 4], [9, 8, 7]]
- C [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
- D [[7, 8, 9], [4, 5, 6], [1, 2, 3]]

Correction : B vaut [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

Question 9

On considère la fonction suivante :

```
def somme(tab):  
    s = 0  
    for i in range(len(tab)):  
        .....  
    return s
```

Par quelle instruction faut-il remplacer les points de suspension pour que l'appel `somme([10, 11, 12, 13, 14])` renvoie 60 ?

Réponses

- A `s = tab[i]`
- B `s = s + tab[i]`
- C `tab[i] = tab[i] + s`
- D `s = s + i`

Correction:

```
def somme(tab):  
    s = 0  
    for i in range(len(tab)):  
        s = s + tab[i]  
    return s
```

```
assert somme([10,11,12,13,14]) == 60
```

Question 10

Qu'affichera le programme suivant ?

```
a = 0  
while a < 3:  
    print("ok")  
    a = a + 1  
print("fini")
```

Correction :

```
ok  
ok  
ok  
fini
```

Question 11

On exécute le script suivant :

```
tab = [('julien',16), ('benoit',17), ('jade',12), ('juliette',17), ('sophie',20)]
```

```
m = tab[0][1]
for personne in tab:
    if (m > personne[1]):
        m = personne[1]
```

Que contient la variable m à la fin de cette exécution ?

Correction : m vaut 12 à la fin de cette exécution.

Question 12

```
phrase = input("Entrer une chaine de caractères ")
indice=0
while indice<len(phrase):
    print(phrase[indice])
    indice = indice + 1
```

Dérouler l'algorithme dans le tableau ci-dessous avec la chaine **"Allan Turing"**

indice													
phrase[indice]													

```
indice= 0
phrase[indice]= A
indice= 1
phrase[indice]= l
indice= 2
phrase[indice]= l
indice= 3
phrase[indice]= a
indice= 4
phrase[indice]= n
indice= 5
phrase[indice]= 
indice= 6
phrase[indice]= T
indice= 7
phrase[indice]= u
```

```
indice= 8  
phrase[indice]= r  
indice= 9  
phrase[indice]= i  
indice= 10  
phrase[indice]= n  
indice= 11  
phrase[indice]= g
```


Question 13

```
compteur = 0
nombre = 13
while nombre > 0 :
    compteur = compteur + 1
    nombre = nombre // 2
print(compteur)
```

Dérouler le programme en complétant le tableau suivant :

nombre							
compteur							
Test							

Quel est l'affichage lorsque ce programme est exécuté ?

Correction :

```
compteur = 0
nombre = 13
while nombre > 0 :
    print("(nombre > 0)=", nombre > 0)
    compteur = compteur + 1
    print("compteur=",compteur)
    nombre = nombre // 2
    print("nombre=",nombre)
print(compteur)
```

```
(nombre > 0)= True
compteur= 1
nombre= 6
(nombre > 0)= True
compteur= 2
nombre= 3
(nombre > 0)= True
compteur= 3
nombre= 1
(nombre > 0)= True
compteur= 4
nombre= 0
4
```