

DS n°1

Nom :

Prénom :

Je devine...

Dans les codes suivants, indiquer, en justifiant brièvement sur l'énoncé, la valeur de la variable X .

```
1 L = [2,1,4,9,10.0]
2 X = L[2] + L[1]
3
4 X = L[ L[1] ]
5
6 X = type(L)
7
8 X = type(L[3])
9
10 X = type(L[4])
11
12 X = len(L)
13
14 X = ( L[0]%2==0 )
15
16 X = 2*L
```

Je corrige...

Dans les codes suivants, proposer une modification pour éviter des erreurs de compilation. Vous pouvez corriger directement sur l'énoncé.

```
1 L = [2,1,2]
2 s = 0
3 for x in L :
4     s+=L[x]
```

```
1 L = [2,1,2]
2 s = 0
3 for k in range(L) :
4     s+=L[k]
```

```
1 N= 10
2 L = []
3 for k in range(N) :
4     L[k] = 0
```

```
1
2 N= 10
3 L = [random() for k in range(N)]
```

```
1 from random import random
2 L = []
3 p = random()
4 if p > 0.5
5     L.append(1)
```

Je code...

Sur une feuille indépendante, rédiger les réponses aux questions suivantes.

a) Exercices

On donne le code suivant :

```
1 c = 0
2 L = [2 , 3 , 5 , 6 , 8 , 10]
3 n = len ( L )
4 for i in range ( n ):
5     if L [ i ] % 2 == 0:
6         c += L [ i ]
7 print ( c )
```

- ☐ **Q 1** - Que fait ce code ?
- ☐ **Q 2** - Que vaut la variable c à la fin de l'exécution du programme ?
- ☐ **Q 3** - Réécrire ce code en effectuant un parcours de liste par éléments et non par indices
- ☐ **Q 4** - Écrire un code permettant de calculer la somme de tous les entiers multiples de 3 ou de 5 inférieurs (strictement) à 1000.
- ☐ **Q 5** - Écrire une fonction `norme(vecteur)` qui prend en entrée un vecteur $[x,y]$ représenté par une liste à deux composantes de flottants et qui renvoie sa norme euclidienne $\sqrt{x^2 + y^2}$. Ne pas oublier d'importer les fonctions mathématiques non natives sous python.

Tournez la page

b) Problème

On rappelle que la fonction `randint(a,b)` de la bibliothèque `random` permet de générer des nombres aléatoires entre `a` et `b` inclus.

□ **Q 6** - Proposer une fonction `Gen_liste(N)` qui renvoie une liste `L` contenant `N` lancers d'un dé à 6 faces

□ **Q 7** - On considère deux listes `L1` et `L2` issues du générateur ci-dessus simulant chaque tour d'un jeu à 2 dés. Ecrire les lignes de codes permettant de générer ces deux listes.

□ **Q 8** - Proposer un code permettant d'obtenir la liste `L3` correspondant à la liste des résultats du jet des 2 dés sur `N` tours de jeu.

□ **Q 9** - Proposer un code permettant d'obtenir le nombre de fois qu'apparaît un double 6 dans `L3`.

□ **Q 10** - Écrire un code permettant de compter le nombre de fois qu'apparaît deux lancers successifs identiques dans `L3`.

□ **Q 11** - Proposer un code permettant d'obtenir la liste `i_8` des indices des lancers pour lesquels le résultat est 8.

□ **Q 12** - En déduire un code permettant de stocker à partir des listes `L1` et `L2` les combinaisons permettant d'obtenir 8.

□ **Q 13** - Les combinaisons se répètent, modifier le code précédent pour ne garder que des combinaisons différentes. On supposera que (4,3) et (3,4) sont deux combinaisons différentes.



Partie 1 : Je devine...

1.1. Valeur de X pour chaque code

1. `L = [2, 1, 4, 9, 10.0]`
`X = L[2] + L[1]`
Réponse : $X = 4 + 1 = 5$
2. `X = L[L[1]]`
Réponse : $X = L[1] = 1$, donc $L[1] = 1$ et $X = L[1] = 1$
3. `X = type(L)`
Réponse : $X = list$, car L est une liste.
4. `X = type(L[3])`
Réponse : $X = int$, car $L[3]$ correspond à 9, qui est un entier.
5. `X = type(L[4])`
Réponse : $X = float$, car $L[4]$ correspond à 10.0 qui est un flottant.
6. `X = len(L)`
Réponse : $X = 5$, car la longueur de la liste est 5.
7. `X = (L[0] % 2 == 0)`
Réponse : $X = True$, car $L[0] = 2$ et $2 \% 2 = 0$.
8. `X = 2 * L`
Réponse : $X = [2, 1, 4, 9, 10.0, 2, 1, 4, 9, 10.0]$, la liste est doublée.

Partie 2 : Je corrige...

2.1.

```
1 L = [2, 1, 2]
2 s = 0
3 for x in range(len(L)): # Utiliser range(len(L)) pour iterer sur les
  indices
4     s += L[x]
```

2.2.

```
1 L = [2, 1, 2]
2 s = 0
3 for k in range(len(L)): # Utiliser range(len(L)) pour iterer sur les
  indices
4     s += L[k]
```

2.3.

```
1 N = 10
2 for k in range(N):
3     L.append(0) # Utiliser append pour remplir
4 L = [0 for _ in range(N)] # Utiliser une comprehension de liste pour
  initialiser
```

2.4.

```
1 from random import random # appeler la fonction
2 L = [random() for k in range(10)]
```

2.5.

```
1 from random import random
2 L = []
3 p = random()
4 if p > 0.5: # ne pas oublier les 2 pts
5     L.append(1)
```

Partie 3 : Je code...

Q1 : Que fait ce code ?

Ce code parcourt la liste L et additionne les éléments pairs de cette liste. La variable c contient cette somme.

Q2 : Valeur de la variable c à la fin de l'exécution

La somme des éléments pairs de la liste $L = [2, 3, 5, 6, 8, 10]$ est $2 + 6 + 8 + 10 = 26$. Donc, à la fin du programme, $c = 26$.

Q3 : Parcours de liste par éléments et non par indices

```
1 c = 0
2 L = [2, 3, 5, 6, 8, 10]
3 for val in L:
4     if val % 2 == 0:
5         c += val
6 print(c)
```

Q4 : Somme des multiples de 3 ou 5 inférieurs à 1000

```
1 s = 0
2 for i in range(1000):
3     if i % 3 == 0 or i % 5 == 0:
4         s += i
5 print(s)
```

Q5 : Fonction pour calculer la norme euclidienne d'un vecteur [x, y]

```

1 from math import sqrt
2
3 def norme(vecteur):
4     x, y = vecteur
5     return sqrt(x**2 + y**2)
6
7 # Exemple d'appel
8 print(norme([3, 4])) # Retourne 5.0

```

Partie 4 : Problème

Q6 : Fonction Gen_liste(N) pour générer N lancers de dés

```

1 from random import randint
2
3 def Gen_liste(N):
4     L = []
5     for _ in range(N):
6         L.append(randint(1, 6))
7     return L

```

Q7 : Générer deux listes simulant les lancers de deux dés

```

1 L1 = Gen_liste(N)
2 L2 = Gen_liste(N)

```

Q8 : Obtenir la liste L3 des résultats des deux dés sur N tours

```

1 L3 = []
2 for i in range(N):
3     L3.append(L1[i] + L2[i])

```

Q9 : Compter le nombre de doubles 6 dans L3

```

1 doubles = 0
2 for i in range(N):
3     if L1[i] == 6 and L2[i] == 6:
4         doubles += 1
5     # ou
6     doubles = 0
7 for i in range(N):
8     if L3[i] == 12:
9         doubles += 1

```

Q10 : Compter les lancers successifs identiques dans L3

```

1 identiques = 0
2 for i in range(1, N):
3     if L3[i] == L3[i - 1]:
4         identiques += 1
5 # ou
6 for i in range(N-1):
7     if L3[i+1] == L3[i]:
8         identiques += 1

```

Q11 : indice des 8

```

1 i_8 = []
2 for i in range(len(L3)):
3     if L3[i] == 8:
4         i_8.append(i)

```

Q12 : comment faire 8

```

1 combinaison = []
2 for i in range(len(i_8)):
3     combinaison.append([ L1[i], L2[i] ])

```

Q13 : comment faire 8 sans doublon

```

1 combinaison_sans_doublon = []
2 for i in range(len(combinaison)):
3     if combinaison[i] not in combinaison_sans_doublon:
4         combinaison_sans_doublon.append([ L1[i], L2[i] ])

```