

ISITCOM

1LM

Examen Principal PYTHON

A.U 2022-2023

<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9
<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9

← Cocher votre numéro CIN ici

Nom , Prénom , Groupe

.....

.....

.....

"If the implementation is hard to explain, it's a bad idea." The zen of python

Les question 1,2,3 et 4 sont dépendentes.

Question 1

Dans cette question, nous allons nous intéresser à la distribution des valeurs des éléments d'une liste d'entiers naturels. Cette distribution représente, pour chaque valeur, le nombre de fois que cette valeur apparaît dans la liste. Le nombre de fois qu'une valeur apparaît est aussi appelé nombre d'occurrences. Par exemple, pour la liste [5,6,7,2,5,1] la distribution est la suivante :

Valeur	0	1	2	3	4	5	6	7
Occurence	0	1	1	0	0	2	1	1

En effet, la valeur 0 apparaît 0 fois, la valeur 1 apparaît 1 fois, la valeur 2 apparaît 1 fois, la valeur 3 apparaît 0 fois, etc.

Ecrivez la fonction **compte_occurrences** qui prend en paramètre, une liste L et un entier x et qui retourne le nombre de fois que la valeur x apparaît dans L.

☐ O ☐ P ☒ Espaces réservé au correcteur

Question 2 Ecrivez la fonction **calcul_distribution** qui prend en paramètre une liste L contenant des entiers et retourne un dictionnaire ayant pour clé les éléments de la liste L sans duplication et comme valeur leurs occurrences .

Exemple :

```

1 L = [1, 5, 6, 7, 2, 5, 1, 5]
2 print(calcul_distribution(L))
3 #Affiche {1:2, 5:3, 6:1, 7:1}

```

☐ O ☐ P ☒ Espaces réservé au correcteur

Question 3

NE RIEN ECRIRE ICI

Question 3 Une distribution de valeur se présente habituellement sous la forme d'un diagramme bâton vertical. Dans un souci de simplicité, nous allons afficher cette distribution à l'aide de bâtons horizontaux formée par des #. Par exemple, la distribution de la liste [5,6,7,2,5,1] s'affichera de la manière suivante :

0 :
1 :#
2 :#
3 :
4 :
5 :##
6 :#
7 :#

Ecrivez la fonction **affiche_distribution** qui prend en paramètre une liste d'entier **L** et affiche leur distribution au format décrit plus haut.

☐ O ☐ P ☒ Espaces réservé au correcteur

Question 4 À partir de la distribution de valeurs, on peut reconstruire la version triée de la liste de départ. Par exemple, avec la la liste $L=[4,1,4,2,1,5,4,7,5]$ qui admet comme distribution $\text{dist}=\{4:3,1:2,2:1,5:2,7:1\}$, la liste triée correspondante est $[1,1,2,4,4,4,5,5,7]$

Ecrivez la fonction `trie_liste` qui prend en paramètre une liste d'entier et retourne la liste triée correspondante en utilisant la fonction `calcul_distribution`. Exemple :

```
1 L = [4, 1, 4, 2, 1, 5, 4, 7, 5]
2 print(trie_liste(L))
3 #Affiche [1, 1, 2, 4, 4, 4, 5, 5, 7]
```

☐ O ☐ P ☒ Espaces réservés au correcteur

Question 5 Donner l'output du code suivant en expliquant la démarche :

```
1 A = [50, 20, 5, 1]
2 n = 137
3 dic = {}
4 for x in A:
5     i = 0
6     while n - x > 0:
7         i += 1
8         n = n - x
9     dic[x] = i
10 print(dic)
```

☐ O ☐ P ☒ Espaces réservés au correcteur

Question 6 Donner l'output du code suivant :

```
1 def mystere(pt, *pts):
2     d = lambda B: (pt[0] - B[0]) ** 2 + (pt[1] - B[1]) ** 2
3     return list(map(d, pts))
4 mystere((0, 0), (0, 1), (3, 4), (6, 8))
```

☐ [5, 1, 20] ☐ [1, 5, 20] ☒ [1, 25, 100] ☐ [0, 0, 0]

Question 7 Les scores finaux d'une compétition ont été enregistrés dans des listes de listes groupe. Pour chaque participant, on a donc une sous-liste [nom, score] avec nom de type string et score de type integer. On suppose que les listes de listes groupe sont non vides, non triées.

Ecrire une fonction **scoreMax** qui renvoie le nom et le score de la personne qui a le plus grand score dans un groupe donné et pour cela vous devez utiliser le paradigme fonctionnelle (Utiliser par exemple des fonctions comme map, filter, reduce ...)
Exemple :

```
1 groupe1 = [ ["Arthur", 34], ["Bernard", 52], ["Jeanne", 18], ["Louise", 45]]
2
3 print(scoreMax(groupe1))
4 # affiche ["Bernard", 52]
```

☐ O ☐ P ☒ Espaces réservés au correcteur

Question 8 On définit le poids d'une chaîne de caractères comme étant la somme des produits des positions de chaque voyelle dans cette chaîne par son rang dans l'alphabet. Si la chaîne ne contient pas de voyelles alors son poids est égal à zéro. Les voyelles sont A, E, I, O, U, Y et leurs rangs respectifs sont : 1, 5, 9, 15, 21, 25. Exemple : La chaîne "BONJOUR" contient 2 fois la voyelle 'O' et une fois la voyelle 'U' : son poids est ainsi égal à $15 \times (2 + 5) + 6 \times 21 = 231$ (2 et 5 sont les positions du 'O', et 6 est la position du 'U').

Corriger les 5 erreurs du programme Python ci-dessous qui calcule et affiche le poids de la chaîne "BONJOUR". Pour chaque erreur, vous préciserez la ligne de l'erreur, si les erreurs de ce programme sont syntaxiques (cas d'une écriture en python incorrecte) ou sémantiques (le programme ne réalise pas ce qui est demandé) ainsi que la correction proposée.

```
1 chaine = "BONJOUR"
2 alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
3 voyelles = "AEIOUY"
4 pos = 0
5 poids = 0
6 for lettre in range(chaine):
7     if lettre in range(voyelles):
8         poids += pos * (alphabet.find(lettre))
9         pos += 1
10 print(poids)
```

☐ O ☐ P ☒ Espaces réservés au correcteur