

1 Python Programming Questions(Loop, Dict, Function)

1. Python program to sort Python Dictionaries by Keys

In [1]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 new_dict={}
3 for i in sorted(a):
4     new_dict[i]=a[i]
5 print(new_dict)
```

```
{1: 10, 20: 15, 23: 62, 25: 32, 50: 70}
```

In [2]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 b = sorted(a.items(),key = lambda x: x[0],reverse=True)
3 print(b)
```

```
[(50, 70), (25, 32), (23, 62), (20, 15), (1, 10)]
```

In [3]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 b = sorted(a.items(),key = lambda x: x[0])
3 print(b)
```

```
[(1, 10), (20, 15), (23, 62), (25, 32), (50, 70)]
```

2. Python program to sort Python Dictionaries by Values

In [4]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 new_dict={}
3 for i in sorted(a.values()):
4     for j in a:
5         if a[j]==i:
6             new_dict[j]=i
7             break
8 print(new_dict)
```

```
{1: 10, 20: 15, 25: 32, 23: 62, 50: 70}
```

In [6]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 b = dict(sorted(a.items(),key=lambda x: x[1]))
3 print(b)
```

```
{1: 10, 20: 15, 25: 32, 23: 62, 50: 70}
```

3. Python program to find the sum of all items in a dictionary

In [8]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 sum_a_item=0
3
4 for i in a.values():
5     sum_a_item+=i
6 print(sum_a_item)
```

189

In [9]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2
3 total = sum(a[i] for i in a)
4 print(total)
```

189

4. Python program to remove a key from a dictionary

In [10]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 del a[23] # del to remove specified key value pair
3 print(a)
```

```
{1: 10, 20: 15, 25: 32, 50: 70}
```

5. Python program to merge two Dictionaries

In [12]:

```
1 a = {1:10, 20:15, 23:62, 25:32, 50:70}
2 b = {5:9, 15:30, 10:22, 6:9, 60:80} # a.update(b)
3                                     # print(a)
4 {**a,**b}
```

Out[12]:

```
{1: 10, 20: 15, 23: 62, 25: 32, 50: 70, 5: 9, 15: 30, 10: 22, 6: 9, 60: 80}
```

6. Program to create grade calculator in Python

In [36]:

```

1 def average(s1):
2     sum1=0
3     for i in s1:
4         sum1+=s1[i]
5     return sum1//len(s1)
6 def grade(average_marks):
7     if average_marks>A:
8         grade1= "Distinction"
9     elif average_marks>B:
10        grade1= "First_Class"
11    elif average_marks>C:
12        grade1= "Second_Class"
13    else:
14        grade1= "Pass_Class"
15    return grade1
16 kiran={"Phy":None,"chem":None,'math':None}
17 Mahesh={"Phy":None,"chem":None,'math':None}
18 Ram={"Phy":None,"chem":None,'math':None}
19 print(" Enter marks of kiran")
20 for i in kiran:
21     kiran[i] = int(input(f"Enter {i} marks"))
22 print(" Enter marks of Mahesh")
23 for i in Mahesh :
24     Mahesh[i] = int(input(f"Enter {i} marks"))
25 print(" Enter marks of Ram")
26 for i in Ram:
27     Ram[i] = int(input(f"Enter {i} marks"))
28
29 A =80
30 B=50
31 C=65
32 average_kiran=average(kiran)
33 average_Mahesh=average(Mahesh)
34 average_Ram=average(Ram)
35 grade_kiran=grade(average_kiran)
36 grade_Mahesh=grade(average_Mahesh)
37 grade_Ram=grade(average_Ram)
38 print("Grade of Mahesh:",grade_Mahesh)
39 print("Grade of kiran:",grade_kiran)
40 print("Grade of Suhas:",grade_Ram)

```

```

Enter marks of kiran
Enter Phy marks97
Enter chem marks98
Enter math marks99
Enter marks of Mahesh
Enter Phy marks65
Enter chem marks70
Enter math marks68
Enter marks of Ram
Enter Phy marks52
Enter chem marks60
Enter math marks55
Grade of Mahesh: First_Class
Grade of kiran: Distinction
Grade of Suhas: First_Class

```

7. Print anagrams together in Python using List and Dictionary

In [31]:

```
1 def anagram_checker(value, words):
2     anagrams = []
3     for word in words:
4         if sorted(word) == sorted(value):
5             anagrams.append(word)
6     return anagrams
7 anagram_checker('cat', ['tac', 'cat', 'pop'])
8
```

Out[31]:

['tac']

In [6]:

```
1 def anagram_check(value, keys):
2     for k in keys:
3         if sorted(value) == sorted(k):
4             return k
5     return None
6 def list_all_anagrams(in_list):
7     dict1 = {}
8     for i in in_list:
9         i = i.lower()
10        key = anagram_check(i, dict1.keys())
11
12        if key:
13            dict1[key].append(i)
14        else:
15            dict1[i] = []
16    return dict1
17 in_list = ['cat', 'hello', 'tiger', 'olleh', 'tac', 'atc', 'regit', 'elephan']
18 list_all_anagrams(in_list)
19
```

Out[6]:

```
{'cat': ['tac', 'atc'], 'hello': ['olleh'], 'tiger': ['regit'], 'elephan':
[]}
```

```
1 8. Check if binary representations of two numbers are an
2 anagram
```

In [7]:

```

1 def check_anagram(num1, num2):
2     flag = False
3     num1 = sorted(num1)
4     num2 = sorted(num2)
5     if num1 == num2:
6         flag = True
7         return flag
8 p = '101010'
9 q = '111010'
10 if check_anagram(p,q):
11     print("Binary representation of two numbers are an anagram")
12 else:
13     print("Binary representation of two numbers are not an anagram")

```

Binary representation of two numbers are not an anagram

```

1 9. Python Counter to find the size of the largest subset of
2 anagram words

```

In [8]:

```

1 def anagram_check(value, keys):
2     for k in keys:
3         if sorted(value) == sorted(k):
4             return k
5     return None
6 def list_all_anagrams(in_list):
7     dict1 = {}
8     in_list1 = in_list.split()
9     for i in in_list1:
10        i = i.lower()
11        key = anagram_check(i,dict1.keys())
12        if key:
13            dict1[key].append(i)
14        else:
15            dict1[i] = []
16    return dict1
17 in_list = "abc pqrstu cba tupqrs gh hg pqtsru"
18 d2 = list_all_anagrams(in_list)
19 print(d2)
20 size = 0
21 for l in d2:
22     if len(d2.get(l)) > size:
23         size = len(d2.get(l))
24         word = l
25 print(f"Largest subset is {word} with length {size}")
26
27

```

```
{'abc': ['cba'], 'pqrstu': ['tupqrs', 'pqtsru'], 'gh': ['hg']}
Largest subset is pqrstu with length 2
```

```

1 10. Python Dictionary to find mirror characters in a
2 string

```

In [9]:

```

1 def mirror_char(x):
2     a = 'abcdefghijklmnopqrstuvwxyz'
3     b = 'zyxwvutsrqponmlkjihgfedcba'
4     f = []
5
6     for i in x:
7         for j in range(len(b)):
8             if i == a[j]:
9                 f.append(b[j])
10    result = ''.join(f)
11    return result
12 x = "python"
13 mirror_char(x)

```

Out[9]:

'kbgslm'

```

1 11. Counting the frequencies in a list using a dictionary in
2 Python

```

In [11]:

```

1 def count_freq(list1):
2     d1 = {}
3     for i in list1:
4         d1[i] = list1.count(i)
5     return d1
6 inp_list = [3,2,1,4,2,1,5,4,5,1,'xyz','q']
7 count_freq(inp_list)

```

Out[11]:

{3: 1, 2: 2, 1: 3, 4: 2, 5: 2, 'xyz': 1, 'q': 1}

```

1 12. Python program to convert a list of Tuples into Dictionary.

```

In [12]:

```

1 def convert_listTuple_dict(list1):
2     dict1 = {}
3     for i, j in list1:
4         dict1[i] = dict1.get(i,[]) + [j]
5     return dict1
6 l1 = [("Tiger", 10), ("Dogs", 12), ("Bear", 14), ("Doves", 20), ("Peacock",25)]
7 print(convert_listTuple_dict(l1))
8
9

```

{ 'Tiger': [10], 'Dogs': [12], 'Bear': [14], 'Doves': [20], 'Peacock': [25]}

```

1 13. Scraping And Finding Ordered Words In A Dictionary
2 using Python

```

In []:

1

```
1 14. Create a list of tuples from the given list having a
2 number and its cube in each tuple
```

In [14]:

```
1 def cube_tuple(list1):
2     result = []
3     result = [(i,i**3)for i in list1]
4     return result
5 l = [7,8,9,3,10,12]
6 cube_tuple(l)
```

Out[14]:

```
[(7, 343), (8, 512), (9, 729), (3, 27), (10, 1000), (12, 1728)]
```

```
1 15. Sort a list of tuples by the second Item
```

In [15]:

```
1 def sort_list_tuple(lis1):
2     list2, list3 = [], []
3     for i in list1:
4         c = (i[::-1])
5         list2.append((c))
6         list2.sort()
7     for j in list2:
8         list3.append(j[::-1])
9     return list3
10 list1 = [(100, 10000), (2, 8), (3, 27), (4, 64), (5, 125), (6, 216),(7,343)]
11 print(sort_list_tuple(list1))
```

```
[(2, 8), (3, 27), (4, 64), (5, 125), (6, 216), (7, 343), (100, 10000)]
```

```
1 16. Python Program for Insertion Sort
```

In [20]:

```
1 def insertion_sort(l1):
2     for i in range(1, len(l1)):
3         key = l1[i]
4         j = i-1
5         while j >=0 and key < l1[j] :
6             l1[j+1] = l1[j]
7             j -= 1
8         l1[j+1] = key
9     return l1
10 l = [300,50,20,70,80,30,-5,3]
11 b = insertion_sort(l)
12 print(b)
13
14
15
16
```

[-5, 3, 20, 30, 50, 70, 80, 300]

1 17. Python Program for SelectionSort

In [23]:

```
1 def selectionSort(array):
2     size = len(array)
3     for j in range(size):
4         k = j
5         for i in range(j + 1, size):
6             if array[i] < array[k]:
7                 k = i
8         (array[j], array[k]) = (array[k], array[j])
9 data = [-5, 55, 1, -11, 7]
10 selectionSort(data)
11 print('Sorted Array in Ascending Order:')
12 print(data)
13
14
```

Sorted Array in Ascending Order:

[-11, -5, 1, 7, 55]

1 18. Python Program for Bubble Sort

In [24]:

```
1 def bubble_sort(list1):
2     for i in range(0,len(list1)-1):
3         for j in range(len(list1)-1):
4             if(list1[j]>list1[j+1]):
5                 temp = list1[j]
6                 list1[j] = list1[j+1]
7                 list1[j+1] = temp
8     return list1
9 list1 = [5, 4, 7, 6, 8, 2]
10 print("The unsorted list is: ", list1)
11 print("The sorted list is: ", bubble_sort(list1))
12
13
14
15
```

The unsorted list is: [5, 4, 7, 6, 8, 2]

The sorted list is: [2, 4, 5, 6, 7, 8]

1 19. Python Program for Merge Sort

In [26]:

```
1 def mergeSort(array):
2     if len(array) > 1:
3
4         r = len(array)//2
5         L = array[:r]
6         M = array[r:]
7
8         mergeSort(L)
9         mergeSort(M)
10        i = j = k = 0
11
12        while i < len(L) and j < len(M):
13            if L[i] < M[j]:
14                array[k] = L[i]
15                i += 1
16            else:
17                array[k] = M[j]
18                j += 1
19            k += 1
20
21        while i < len(L):
22            array[k] = L[i]
23            i += 1
24            k += 1
25        while j < len(M):
26            array[k] = M[j]
27            j += 1
28            k += 1
29
30 def printList(array):
31     for i in range(len(array)):
32         print(array[i], end=" ")
33     print()
34
35 array = [10,2,3,5,7,9]
36 mergeSort(array)
37 print("Sorted array is: ")
38 printList(array)
```

Sorted array is:

2 3 5 7 9 10

```
1 20. Python Program for QuickSortSort
2
```

In [29]:

```
1 def prepare(numbers, min, max):
2     pivot = numbers[max]
3     item = min - 1
4     for i in range(min, max):
5         if numbers[i] <= pivot:
6             item = item + 1
7             (numbers[item], numbers[i]) = (numbers[i], numbers[item])
8     (numbers[item + 1], numbers[max]) = (numbers[max], numbers[item + 1])
9     return item + 1
10 def quick_sort(nums, min, max):
11     if min < max:
12         k = prepare(nums, min, max)
13         quick_sort(nums, min, k-1)
14         quick_sort(nums, k+1, max)
15 nums = [11, 7, 9, 4, 3, 5, -10]
16 size = len(nums)
17 quick_sort(nums, 0, size - 1)
18 print(nums)
```

[-10, 3, 4, 5, 7, 9, 11]

In []:

1