

1. Consider two disjoint lists with each having 10 elements. Now, randomly generate the number and add it to the list alternatively. Stop the process when intersection of both the lists is a singleton set. The range of the values in the lists is [0,100]. Calculate the total number of distinct elements in both the lists.

```
[ ]: a=[1,2,3,4,5,11,12,13,14,15]
b=[6,7,8,9,10,16,17,18,19,20]
a=set(a)
b=set(b)
c=1
import random
while len(a.intersection(b))!=1:
    if c%2!=0:
        a.add(random.randint(0,101))
    else:
        b.add(random.randint(0,101))
    c+=1
print(len(a),len(b))
print(len(a.symmetric_difference(b)))
```

19 17

34

Q2. There is an array of n integers. There are also 2 disjoint sets, A and B, each containing m integers. You like all the integers in set A and dislike all the integers in set B. Your initial happiness is 0. For each i integer in the array, if i (in A , you add 1 to your happiness. If i (in B, you add -1 to your happiness. Otherwise, your happiness does not change. Output your final happiness at the end. Note: Since A and B are sets, they have no repeated elements. However, the array might contain duplicate elements.

```
[ ]: m=int(input("Enter how many distinct integers you want to enter : "))
a=set()
b=set()
for j in range(m):
    a.add(input('SET A INTEGER '))
    b.add(input("SET B INTEGER "))
n=int(input("Enter the n integers for the array :"))
print(a)
print(b)
happiness=0
for i in range(1,n+1):
    if i in a:
        happiness+=1
    if i in b:
        happiness-=1
print("Total Happiness :",happiness)
```

Enter how many distinct integers you want to enter : 3

SET A INTEGER 1

SET B INTEGER 2

SET A INTEGER 3

SET B INTEGER 4

SET A INTEGER 5

SET B INTEGER 6

Enter the n integers for the array :7

{'1', '5', '3'}

{'4', '2', '6'}

Total Happiness : 0

3. Students of DAIICT have subscriptions to English and Hindi magazines. Some students have subscribed to English only, some have subscribed to Hindi only, and some have subscribed to both magazines. You are given two sets of student roll numbers. One set has subscribed to the English magazines, and one set has subscribed to the Hindi magazines. Your task is to find the total number of students who have subscribed to either the English or the Hindi magazines but not both.

```
[ ]: a=int(input("Enter total number of those students who have subscribed for English_
↳magazine :"))
b=int(input("Enter total number of those students who have subscribed for Hindi_
↳magazine :"))
e=[]
h=[]

for i in range(a):
    r=int(input("Enter rollnumber for English magazine :"))
    e.append(r)

for j in range(b):
    p=int(input("Enter rollnumber for Hindi magazine :"))
    h.append(p)
print("English magazine roll numbers :",e)
print("Hindi magazine roll numbers :",h)
z=set(e).union(set(h))

print("Total number of student subscribed either the English or the Hindi magazines :
↳",len(set(z)))
```

Enter total number of those students who have subscribed for English magazine :4

Enter total number of those students who have subscribed for Hindi magazine :4

Enter rollnumber for English magazine :101

Enter rollnumber for English magazine :102

Enter rollnumber for English magazine :103

Enter rollnumber for English magazine :104

Enter rollnumber for Hindi magazine :103

Enter rollnumber for Hindi magazine :104

Enter rollnumber for Hindi magazine :105

Enter rollnumber for Hindi magazine :106

English magazine roll numbers : [101, 102, 103, 104]

Hindi magazine roll numbers : [103, 104, 105, 106]

Total number of student subscribed either the English or the Hindi magazines : 6

4. Given n tuples of n people details, formatted as (name, phone number). Now, it is assumed that each family is having only one phone number. Now identify the number of families with family size >4, 4, 3, 2, 1. Here, consider n>20.

```
[7]: A=()
A=list(A)
```

```

A=[]
n=int(input("enter n>20 : "))
for i in range(n):
    var = input("Enter name : ")
    num = eval(input("enter phone number : "))
    tup=(var,num)
    A.append(tup)
print(A)

dict1={}
for x in A:
    if x[1] in dict1:
        dict1[x[1]]+=1
    else:
        dict1[x[1]]=1
print(dict1)
print(dict1.items())
dict2={">4":0,4:0,3:0,2:0,1:0}
for i in dict1.items():
    if i[1]>4:
        dict2[">4"]+=1
    elif i[1]==4:
        dict2[4]+=1
    elif i[1]==3:
        dict2[3]+=1
    elif i[1]==2:
        dict2[2]+=1
    else:
        dict2[1]+=1
print(dict2)

```

```

enter n>20 : 21
Enter name : a
enter phone number : 101
Enter name : b
enter phone number : 102
Enter name : c
enter phone number : 103
Enter name : d
enter phone number : 104
Enter name : e
enter phone number : 101
Enter name : f
enter phone number : 102
Enter name : g
enter phone number : 103
Enter name : h
enter phone number : 104
Enter name : i
enter phone number : 109
Enter name : j
enter phone number : 109

```

```

Enter name : k
enter phone number : 102
Enter name : l
enter phone number : 102
Enter name : m
enter phone number : 104
Enter name : n
enter phone number : 106
Enter name : o
enter phone number : 109
Enter name : p
enter phone number : 104
Enter name : q
enter phone number : 105
Enter name : r
enter phone number : 106
Enter name : s
enter phone number : 107
Enter name : t
enter phone number : 104
Enter name : u
enter phone number : 106
[('a', 101), ('b', 102), ('c', 103), ('d', 104), ('e', 101), ('f', 102), ('g', 103),
('h', 104), ('i', 109), ('j', 109), ('k', 102), ('l', 102), ('m', 104), ('n', 106), ('o',
109), ('p', 104), ('q', 105), ('r', 106), ('s', 107), ('t', 104), ('u', 106)]
{101: 2, 102: 4, 103: 2, 104: 5, 109: 3, 106: 3, 105: 1, 107: 1}
dict_items([(101, 2), (102, 4), (103, 2), (104, 5), (109, 3), (106, 3), (105, 1), (107,
1)])
{'>4': 1, 4: 1, 3: 2, 2: 2, 1: 2}

```

5. Write a function histogram() to build a histogram that takes a string and builds a frequency listing of the characters contained in it.

```

[ ]: def histogram(string):
    dict1={}
    lis=list(string)
    for i in lis:
        if i not in dict1:
            dict1[i]=1
        else:
            dict1[i]+=1
    print(dict1)

string="To tom mot rotm rjm rt"
histogram(string)

```

```
{'T': 1, 'o': 4, ' ': 5, 't': 4, 'm': 4, 'r': 3, 'j': 1}
```

```
[ ]:
```

6. Consider character list i.e. the list that contains only characters. Now this list can have same character multiple times and also contains small-cased and upper-cased alphabets. Identify the missing characters

(if any) in this character list, so that, inclusion of those missing characters, will complete all the upper-cased alphabets in that given list. Note: This needs to be done without looping

```
[ ]: n=input("Enter string :")
m=list(n)
list1=set(m)
list2=[]
for i in range(ord("A"),ord("Z")+1):
    list2.append(chr(i))
for j in range(ord("a"),ord("z")+1):
    list2.append(chr(j))
s=set(list2)
X=s - list1
print(X)
```

Enter string :ALJKHDSLGKHAOQEIUTYOEQMZXVBlakdshgaklqoiueytweoiquqpmxzvnbadskhlghdalzm
{'P', 'j', 'F', 'f', 'C', 'r', 'R', 'c', 'N'}

7. Remove all the duplicate words in a given sentence.

```
[ ]: n=input("Enter sentence with duplicates words :")
lis=n.split()
liss=[]
for i in lis:
    if ((i not in liss )):
        liss.append(i)
print(' '.join(liss))
```

Enter sentence with duplicates words :Ram is singer and shyam is also singer
Ram is singer and shyam also

```
[ ]: #2nd way
n=input("Enter sentence with duplicates words :")
sp=n.lower().split()
print(sp)
s=set(sp)
p=list(s)
print(" ".join(p))
```

Enter sentence with duplicates words :Ram is singer and shyam is also singer
['ram', 'is', 'singer', 'and', 'shyam', 'is', 'also', 'singer']
shyam also is singer ram and

```
[ ]:
```

8. Give a dictionary with value lists, sort the keys by summation of values in value list. Input : test_dict = {'ABCD' : [2,3,3], 'PQRS' : [1,1,3]} Output : {'PQRS': 5, 'ABCD': 8}

```
[ ]: test = {"ABCD": [2,3,3], "PQRS": [1,1,3]}
#print(test.items)
for i in test.items():

    #print(i[0])
```

```
#print(i[1])
test[i[0]]=sum(i[1])
print(test)
print(dict(sorted(test.items(),key=lambda i:i[1])))
```

```
{'ABCD': 8, 'PQRS': 5}
{'PQRS': 5, 'ABCD': 8}
```

[]:

9. Prepare a list-dictionary i.e. a dictionary with its values given as a list. Now, extract the unique elements from the dictionary values.

```
[ ]: test = {"ABCD": [2,3,3], "PQRS": [1,1,3]}
lis=[]
for i in test.items():
    lis.extend(i[1])
print(set(lis))
```

```
{1, 2, 3}
```

10. Ask the user to enter n entries. Now start grouping the entered numbers based on its sum of the digits. Now find the group which is having the highest number of entries in it.

```
[ ]: n= int(input("enter how many entries you want "))
dict1={}
for i in range(n):
    m=int(input("entry : "))
    temp=m
    sum=0
    while( temp>0):
        rem=temp%10
        sum+=rem
        temp=temp//10
    print("sum :",sum)
    if sum in dict1:
        dict1[sum].append(m)
    else:
        dict1[sum]=[m]
print(dict1)
print(sorted(dict1.items(),key=lambda i:len(i[1]))[-1])
```

```
enter how many entries you want 5
entry : 150
sum : 6
entry : 240
sum : 6
entry : 111
sum : 3
entry : 120
sum : 3
entry : 158
sum : 14
```

```
{6: [150, 240], 3: [111, 120], 14: [158]}  
(3, [111, 120])
```

```
[ ]: 2
```

```
[ ]: 2
```

```
[ ]:
```