**Ex. No. : 5.5 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Count Chars**

Write a python program to count all letters, digits, and special symbols respectively from a given string

**For example:**

| **Input** | **Result** |
| --- | --- |
| rec@123 | 3  3  1 |

**Program:**

x=input()

a,b,c=0,0,0

for i in x:

if(i.isalpha()):

a+=1

elif(i.isalnum()):

b+=1

else:

c+=1

print(a,b,c,sep="\n")



**Ex. No. : 5.6 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Reverse String**

Reverse a string without affecting special characters. Given a string S, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

Input:

A&B

Output:

B&A

Explanation: As we ignore '&' and

As we ignore '&' and then reverse, so answer is "B&A".

For example:

Input Result

A&x#

x&A#

**Program:**

s=input()

l=[]

for i in s:

if(i.isalpha()):

l.append(i)

l.reverse()

r=''

index=0

for i in s:

if(i.isalpha()):

r+=l[index]

index+=1

else:

r+=i

print(r)

****

**Ex. No. : 5.7 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Longest Word**

Write a python to read a sentence and print its longest word and its length

**For example:**

| **Input** | **Result** |
| --- | --- |
| This is a sample text to test | sample  6 |

**Program:**

sen=input()

words=sen.split()

l=""

maxi=0

for word in words:

if(len(word)>maxi):

l=word

maxi=len(word)

print(l,maxi,sep="\n")



**Ex. No. : 5.8 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Remove Palindrome Words**

String should contain only the words are not palindrome.

Sample Input 1

Malayalam is my mother tongue

Sample Output 1

is my mother tongue

**Program:**

s=input()

words=s.split()

x=''

for word in words:

word=word.lower()

if (word!=word[::-1]):

print(word,end=" ")

****

**Ex. No. : 5.9 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Remove Characters**

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

experience

enc

Sample Output 1

xpri

**Program:**

s1=input()

s2=input()

x=''.join(char for char in s1 if char not in s2)

print(x)

****

**Ex. No. : 5.10 Date: 17.04.24**

**Register No.: 231801123 Name: Pavithra S**

**Unique Names**

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

**Input:**

first

second

first

third

second

then your program should display:

**Output:**

first

second

third

**Program:**

l=[]

while(True):

a=input()

if a!=" ":

l.append(a)

else:

break

l=dict.fromkeys(l)

for i in l:

print(i)

****

### [**06 -** **List in Python**](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84#section-5)

**Ex. No. : 6.1 Date: 04.05.24**

**Register No.:231801123 Name: Pavithra S**

**Element Insertion**

Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

Sample Test Cases

Test Case 1

Input

1

3

4

5

6

7

8

9

10

11

2

Output

ITEM to be inserted:2

After insertion array is:

1

2

3

4

5

6

7

8

9

10

11

Test Case 2

Input

11

22

33

55

66

77

88

99

110

120

44

Output

ITEM to be inserted:44

After insertion array is:

11

22

33

44

55

66

77

88

99

110

120

**Program:**

x=[]

for i in range(0,11):

b=int(input())

x.append(b)

#a.sort()

print("ITEM to be inserted:",x[-1],sep='')

x.sort()

print("After insertion array is:")

for i in x:

print(i)



**Ex. No. : 6.2 Date: 04.05.24**

**Register No.:231801123 Name: Pavithra S**

**Anagram**

Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A.

We want to find an *index mapping* P, from A to B. A mapping P[i] = j means the ith element in A appears in B at index j.

These lists A and B may contain duplicates. If there are multiple answers, output any of them.

For example, given

**Input**

5

12 28 46 32 50

50 12 32 46 28

**Output**

1 4 3 2 0

**Explanation**

A = [12, 28, 46, 32, 50]

B = [50, 12, 32, 46, 28]

We should return

[1, 4, 3, 2, 0]

as P[0] = 1 because the 0th element of A appears at B[1], and P[1] = 4 because the 1st element of A appears at B[4], and so on.

**Note:**

1. A, B have equal lengths in range [1, 100].
2. A[i], B[i] are integers in range [0, 10^5].

**Program:**

def index\_mapping(A, B):

index\_map = {num: i for i, num in enumerate(B)}

return ' '.join(str(index\_map[num]) for num in A)

n=int(input())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

print(index\_mapping(A, B))

****

**Ex. No. : 6.3 Date: 04.05.24**

**Register No.: 231801123 Name: Pavithra S**

**Merge Two Sorted Arrays Without Duplication**

Output is a merged array without duplicates.

Input Format

N1 - no of elements in array 1

Array elements for array 1

N2 - no of elements in array 2

Array elements for array2

Output Format

Display the merged array

**Sample Input 1**

5

1

2

3

6

9

4

2

4

5

10

**Sample Output 1**

1 2 3 4 5 6 9 10

**Program:**

n1=int(input())

l1=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

l2=[]

for i in range(0,n2):

a=int(input())

l2.append(a)

l3=[]

l3.extend(l1)

l3.extend(l2)

a=list(set(l3))

a.sort()

for i in a:

print(i,end=' ') n1=int(input())

l1=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

l2=[]

for i in range(0,n2):

a=int(input())

l2.append(a)

l3=[]

l3.extend(l1)

l3.extend(l2)

a=list(set(l3))

a.sort()

for i in a:

print(i,end=' ')

****

**Ex. No. : 6.4 Date: 04.05.24**

**Register No.: 231801123 Name: Pavithra S**

**Distinct Elements in an Array**

Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

Example Input:

5

1

2

2

3

4

Output:

1 2 3 4

Example Input:

6

1

1

2

2

3

3

Output:

1 2 3

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  1  2  2  3  4 | 1 2 3 4 |
| 6  1  1  2  2  3  3 | 1 2 3 |

**Program:**

n = int(input())

arr = []

for \_ in range(n):

arr.append(int(input()))

distinct\_elements = set(arr)

print(\*distinct\_elements)

****

**Ex. No. : 6.5 Date: 04.05.24**

**Register No.: 231801123 Name: Pavithra S**

**The Pivot**

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

arr=[1,2,3,4,6]

·         the sum of the first three elements, 1+2+3=6. The value of the last element is 6.

·         Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.

·         The index of the pivot is 3.

Constraints

·         3 ≤ n ≤ 105

·         1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ i < n

·         It is guaranteed that a solution always exists.

The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer, arr[i], where 0 ≤ i < n.

Sample Case 0

Sample Input 0

4

1

2

3

3

Sample Output 0

2

Explanation 0

·         The sum of the first two elements, 1+2=3. The value of the last element is 3.

·         Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.

·         The index of the pivot is 2.

Sample Case 1

Sample Input 1

3

1

2

1

Sample Output 1

1

Explanation 1

·         The first and last elements are equal to 1.

·         Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.

·         The index of the pivot is 1.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  1  2  3  3 | 2 |
| 3  1  2  1 | 1 |

**Program:**

a = int(input())

b= []

for i in range(a):

element = int(input())

b.append(element)

total= sum(b)

left= 0

right = total- b[0]

if left== right:

print(0)

exit()

for i in range(1, a):

left+= b[i - 1]

right-= b[i]

if left== right:

print(i)

break

****

**Ex. No. : 6.6 Date: 04.05.24**

**Register No.: 231801123 Name: Pavithra S**

**Intersection of array**

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

The first line contains T, the number of test cases. Following T lines contain:

1.      Line 1 contains N1, followed by N1 integers of the first array

2.      Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

7

1

2

3

3

4

5

6

2

1

6

Output:

1 6

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1  3  10  17  57  6  2  7  10  15  57  246 | 10 57 |
| 1  7  1  2  3  3  4  5  6  2  1  6 | 1 6 |

**Program:**

t=int(input())

l1=list()

while(t!=0):

n1=int(input())

l1=[]

l2=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

for i in range(0,n2):

a=int(input())

l2.append(a)

t=t-1

c=set(l1)

d=set(l2)

e=list(c.intersection(d))

e.sort()

for i in e:

print(i,end=' ')

print('\n')

****