L9 : Arrays and Lists Practice Questions

1-Tut : Predict the Output

Send Feedback

What will be the output of the following code?

```
li =['abcd','def']
li.insert(4,5)
print(li)
```

Note : li.insert(index,value) : if index out of range default it will be added after the last available index

Options

Index Error
['abcd', 'def']
['abcd', 'def', 5]
None of the above

Correct Answer : C

2-Tut : Predict the Output

Send Feedback

What will be the output of the following code?

```
li = ['abcd',5,'def',5]
li.remove(5)

print(li)
```

Note : remove(5) will remove only the first occurrence of 5 from the list

Options

Error
['abcd', 5, 'def']
['abcd', 'def']
['abcd', 'def', 5]

Correct Answer : D

3-Tut : Predict the Output

Send Feedback

What will be the output of the following code?

```
li = [5,2,6,8]
li.pop(2)

print(li)
```

Options

[5, 6, 8]
[5, 2, 8]
[5, 2, 6, 8]
Error

Correct Answer : B

Solution Description

The pop() method returns the item present at the given index.

Note : some more operations on list : 1. Slicing of a list, 2. li.append(), 3.li.extend()

4-Tut : Predict the Output

Send Feedback

What will be the output of the following code?

```
li = [1,2,3,4,5]
for i in li[1:4]:
    print(i,end= " ")
```

Options

1 2 3 4 5
2 3 4 5
2 3 4
None of the above

Correct Answer : C

Negative Indexing:(index start from -1 till -length , -1 represents last index, -length is for 0th index) and Sequencing (list [start : end : step] start is included end is excluded)

Line separated input of list

5-Tut : Predict the Output

Send Feedback

What will be the output of code if the following input is provided?

```
5
1
2
3
4
5
n = int(input())
li = []
for i in range(n):
    li.append(input())

print(li)
```

Options

[1, 2, 3, 4, 5]
['1', '2', '3', '4', '5']
['12345']
None of the above

Correct Answer : B

Space Separated Input of list : input() function by default deal with string, string.split(delimiter) is used to split string into list on basis of a specified delimited : (default delimiter is : ' ' space) there are lots of other ways to take space separated input for list but python have something amazing that we can do everything in just single line of code

Ex : li = [int(x) for x in input().split()]

6-Tut : Predict the Output

Send Feedback
What will be the output of code if the following input is provided?

1 3 6 8 9

```
li = [x for x in input().split()]

print(li)
```

Options

['1', '3', '6', '8', '9']
[1, 3, 6, 8, 9]
['1 3 6 8 9']
None of the above

Correct Answer : A

7-Tut : Array Sum

Send Feedback
Given an array of length N, you need to find and print the sum of all elements of the array.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Output Format : Sum

Constraints :
1 <= N <= 10^6

Sample Input : 3

9 8 9

Sample Output : 26

```
## Read input as specified in the question.
## Print output as specified in the question.
N = int(input())
list = [int(x) for x in input().split()]
sum = 0
for ele in list:
    sum = sum + ele
print(sum)
```

8-Tut : Predict the Output

Send Feedback
What will be the output of the following code?

```
def change(li):

  li[1] = li[1] + 2

li = [1,2,3,4,5]

change(li)

print(li)
```

Options

[3, 2, 3, 4, 5]
[1, 4, 3, 4, 5]
[1, 2, 3, 4, 5]
None of the above

Correct Answer : B

9-Tut : Predict the Output

Send Feedback
What will be the output of the following code?

```
def change(li):

    li[1] = li[1] + 2

    li = [3,3,3,4,5]

li = [1,2,3,4,5]

change(li)

print(li)
```

Options

[3, 2, 3, 4, 5]
[1, 4, 3, 4, 5]
[3, 3, 3, 4, 5]
None of the above

Correct Answer : B

Reversing a list in python :

```
for i in range(len(list)):

    list [i] , list[ length- i - 1 ] = list [ length - i - 1 ], list[ i ]

    # method 2 : using negative indexing : li [ i ], li [ -i - 1 ] = li [ -i - 1 ], li [ i ]
```

Swap in python in 1 line : a,b = b,a

Reverse list without loop, in 1 line in python using Slicing Concept :

```
List = List [ : : -1 ]
```

10-Tut : Swap Alternate

Send Feedback

You have been given an array/list(ARR) of size N. You need to swap every pair of alternate elements in the array/list.

You don't need to print or return anything, just change in the input array itself.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format :

For each test case, print the elements of the resulting array in a single row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

Time Limit: 1sec

Sample Input 1:

1

6

9 3 6 12 4 32

Sample Output 1 :

3 9 12 6 32 4

Sample Input 2:

2

9

9 3 6 12 4 32 5 11 19

4

1 2 3 4

Sample Output 2 :

3 9 12 6 32 4 11 5 19

2 1 4 3

```
def swapAlternate(arr, n) :
    #Your code goes here
    if n % 2 == 0:
        for i in range(0,n,2):
            arr[i],arr[i+1] = arr[i+1],arr[i]
    else:
        for i in range(0,n-1,2):
            arr[i],arr[i+1] = arr[i+1],arr[i]
```

11-Ass : Find Unique

Send Feedback

You have been given an integer array/list(ARR) of size N. Where N is equal to [2M + 1].
Now, in the given array/list, 'M' numbers are present twice and one number is present only once.
You need to find and return that number which is unique in the array/list.

Note:

Unique element is always present in the array/list according to the given condition.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format : For each test case, print the unique element present in the array.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^3

Time Limit: 1 sec

Sample Input 1:

1

7

2 3 1 6 3 6 2

Sample Output 1: 1
Sample Input 2:

2

5

2 4 7 2 7

9

1 3 1 3 6 6 7 10 7

Sample Output 2:

4

10

Method : 1 : TC **:** O(n*n) : pick each element one by one and check its occurrence

Method : 2 : TC : O(n): take the benefit of properties of XOR

```
def findUnique(arr, n) :
    #Your code goes here
    uniq = 0
    for ele in arr:
        uniq = uniq ^ ele
    return uniq
```

12-Ass : Find Duplicate

Send Feedback

You have been given an integer array/list(ARR) of size N which contains numbers from 0 to (N - 2). Each number is present at least once. That is, if N = 5, the array/list constitutes values ranging from 0 to 3 and among these, there is a single integer value that is present twice. You need to find and return that duplicate number present in the array.

Note : Duplicate number is always present in the given array/list.
Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Output Format : For each test case, print the duplicate element in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^3

Time Limit: 1 sec

Sample Input 1:

1

9

0 7 2 5 4 7 1 3 6

Sample Output 1:

7

Sample Input 2:

2

5

0 2 1 3 1

7

0 3 1 5 4 3 2

Sample Output 2:

1

3

```
def duplicateNumber(arr, n) :
    #Your code goes here
    for i in range(0,n-1,1):
        for j in range(i+1,n,1):
            if(arr[i] == arr[j]):
                return arr[i]
```

13-Ass : Array Intersection

Send Feedback

You have been given two integer arrays/list (ARR1 and ARR2) of size N and M, respectively. You need to print their intersection; An intersection for this problem can be defined when both the arrays/lists contain a particular value or to put it in other words, when there is a common value that exists in both the arrays/lists.

Note : Input arrays/lists can contain duplicate elements.

The intersection elements printed would be in the order they appear in the first array/list(ARR1)

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array/list.

Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output format :

For each test case, print the intersection elements in a row, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

0 <= M <= 10^5

Time Limit: 1 sec

Sample Input 1 :

2

6

2 6 8 5 4 3

4

2 3 4 7

2

10 10

1

10

Sample Output 1 :

2 4 3

10

Sample Input 2 :

1

4

2 6 1 2

5

1 2 3 4 2

Sample Output 2 :

2 1 2

Explanation for Sample Output 2 :

Since both input arrays have two '2's, the intersection of the arrays also have two '2's. The first '2' of the first array matches with the first '2' of the second array. Similarly, the second '2' of the first array matches with the second '2' if the second array.

```
def intersections(arr1, n, arr2, m) :
    #Your code goes here
    for i in range(0,n,1):
        for j in range(0,m,1):
            if (arr1[i] == arr2[j]):
                print(arr1[i],end=" ")
                arr2[j] = -1000000007
                break
```

14-Ass : Pair Sum

Send Feedback

You have been given an integer array/list(ARR) and a number X. Find and return the total number of pairs in the array/list which sum to X.

Note: Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains an integer 'X'.

Output format :

For each test case, print the total number of pairs present in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^3

0 <= X <= 10^9

Time Limit: 1 sec

Sample Input 1:

1

9

1 3 6 2 5 4 3 2 4

7

Sample Output 1: 7
Sample Input 2:

2

9

1 3 6 2 5 4 3 2 4

12

6

2 8 10 5 -2 5

10

Sample Output 2:

0

2

Explanation for Input 2:

Since there doesn't exist any pair with a sum equal to 12 for the first query, we print 0.

For the second query, we have 2 pairs in total that sum up to 10. They are, (2, 8) and (5, 5).

```
def pairSum(arr, n, x) :
    #Your code goes here
    count = 0
    for i in range(0,n-1,1):
        for j in range(i+1,n,1):
            if(arr[i] + arr[j] == x):
                count += 1
    return count
```

15-Ass : Triplet Sum

Send Feedback

You have been given a random integer array/list(ARR) and a number X. Find and return the number of triplets in the array/list which sum to X.

Note :

Given array/list can contain duplicate elements.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains an integer 'X'.

Output format :

For each test case, print the total number of triplets present in the array/list.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 50

0 <= N <= 10^2

0 <= X <= 10^9

Time Limit: 1 sec

Sample Input 1:

1

7

1 2 3 4 5 6 7

12

Sample Output 1: 5

Sample Input 2:

2

7

1 2 3 4 5 6 7

19

9

2 -5 8 -6 0 5 10 11 -3

10

Sample Output 2:

0

5

Explanation for Input 2:

Since there doesn't exist any triplet with sum equal to 19 for the first query, we print 0.

For the second query, we have 5 triplets in total that sum up to 10. They are, (2, 8, 0), (2, 11, -3), (-5, 5, 10), (8, 5, -3) and (-6, 5, 11)

```
def findTriplet(arr, n, x) :
    #Your code goes here
    count = 0
    for i in range(0,n-2,1):
        for j in range(i+1,n-1,1):
            for k in range(j+1,n,1):
                if (arr[i] + arr[j] + arr[k] == x):
                    count += 1
    return count
```

Note : Above time complexity is cubic : O(n^3) : will see other optimised version later

16-Ass : Sort 0 1

Send Feedback

You have been given an integer array/list(ARR) of size N that contains only integers, 0 and 1. Write a function to sort this array/list. Think of a solution which scans the array/list only once and don't require use of an extra array/list.

Note:

You need to change the given array/list itself. Hence, no need to return or print anything.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers(all 0s and 1s) representing the elements in the array/list.

Output format :

For each test case, print the sorted array/list elements in a row separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^5

Time Limit: 1 sec

Sample Input 1:

1

7

0 1 1 0 1 0 1

Sample Output 1:

0 0 0 1 1 1 1

Sample Input 2:

2

8

1 0 1 1 0 1 0 1

5

0 1 0 1 0

Sample Output 2:

0 0 0 1 1 1 1 1

0 0 0 1 1

```
def sortZeroesAndOne(arr, n) :
    #Your code goes here
    i = 0
    j = n-1
    while(i <= j):
        if(arr[i] == 0):
            i += 1
        elif(arr[j] == 0):
            arr[i],arr[j] = arr[j],arr[i]
            i += 1
            j -= 1
        else:
            j -= 1
```