Lecture1 Arrays - 1_PROBLEMS

Correct Statement

Send Feedback

Figure out the correct statement for the below line of code ?

```
int[] arr = new int[5];
arr = new int[6];
```

Options

Attempts left: 0/2

This problem has only one correct answer

- The code has compile error because the variable arr cannot be changed once it is assigned.
- The code will compile and run fine. The second line assigns a new array to arr.
- The code has compile errors because we cannot assign a different size array to arr.
- None of these

Hurray! Correct Answer

Predict The Output

Send Feedback

What will be the output of the following code ?

```
int arr[] = new int[5];
System.out.println(arr[0]);
```

Options

This problem has only one correct answer

- Garbage value
- ArrayIndexOutOfBoundsException
- 0

Hurray! Correct Answer

Solution Description

####Whenever an integer array is created, all elements are initialised to 0.

Predict The Output

Send Feedback

What will be the output of the following code ?

```
char chArray[] = new char[15];
System.out.println(chArray[15]);
```

Options

Attempts left: 1/2

This problem has only one correct answer

- Garbage value
- ArrayIndexOutOfBoundsException
- 0

Hurray! Correct Answer

Solution Description

For an array of size n, indexes of the array range from 0 to (n - 1). So here for array of size 15, valid indices are from 0 to 14. This chArray[15] is trying to access an invalid index which gives ArrayIndexOutOfBoundsException.

What will be the output of the following code ?

```
boolean arr[] = new boolean[5];
System.out.println(arr[0]);
```

false ✔

Correct Answer

Solution Description

####Whenever an boolean array is created, all elements are initialised to false.

Predict The Output

Send Feedback

What will be the output ?

```
public class Main {
   public static void main(String args[]){
     int arr[] = new int[5];
     arr[1] = 10;
     arr = new int[2];
     System.out.println(arr[0]);
   }
}
```

Options

Attempts left: 1/2

This problem has only one correct answer

- ○ 10
- ◉ 0
- ○ Compilation Error

✓ Hurray! Correct Answer

Solution Description

####arr initially refers to an array of size 5, line 2 assigns 10 to arr[1] but in line 3 arr is made to point to a new array of size 2. By default all elements of integer array are initialized to 0. Hence line 4 will print 0.

Length Of Array

Send Feedback

Which of the options can be used to get the size of an array?

***arr is the array.**

Options

Attempts left: 1/2

This problem has only one correct answer

- ○ arr.size
- ◉ arr.length
- ○ lengthof(arr)
- ○ None of these

✓ Hurray! Correct Answer

Non Primitives

Send Feedback

Which among the following are non-primitive datatypes?

Options

Attempts left: 1/2

This problem may have one or more correct answers

- ☑ Array ✓
- ☐ int
- ☐ char
- ☑ Scanner ✓

✓ Hurray! Correct Answer

Predict The Output

Send Feedback

What will be the output of the following code?

```
public static int sum(int [] arr)
{
   int arrsum=0;
   for(int i=0;i<5;i++)
   arrsum+=arr[i];
   return arrsum;
}
public static void main (String[] args) {
int arr[]={1,2,3,4,5,6,7,8};
System.out.print(sum(arr));
}
```

Options

Attempts left: 0/2

This problem has only one correct answer

- 36
- 15 ✓
- 21
- 10

Hurray! Correct Answer

Predict The Output

Send Feedback

What will be the output of the following code?

```
public static void mul(int [] arr)
{
   for(int i=0;i<5;i++)
   arr[i]*=i;

}
public static void main (String[] args) {
   int arr[]={1,2,3,4,5};
   mul(arr);
   for(int i=0;i<5;i++)
   {
      System.out.print(arr[i]);
   }
}
```

Options

Attempts left: 1/2

This problem has only one correct answer

- 12345
- 1491625
- 16122030
- 0261220 ✓

Hurray! Correct Answer

Predict The Output

Send Feedback

What will be the output of the following code ?

```
public class Main {
   public static void change(int input[]){
   input[0] = 15;
   }

   public static void main(String args[]){
      int arr[] = new int[5];
      change(arr);
      System.out.println(arr[0]);
   }
}
```

Options

Attempts left: 1/2

This problem has only one correct answer

- 15 ✓
- 0
- Error

Hurray! Correct Answer

Solution Description

####"arr" is a reference to the array, which contains address of the array. Hence we have passed this address to function "change". Thus arr and input both refer to the same array. Hence the statement input[0] = 15 changes the value of element at index 0 of the array to 15.

Predict The Output

Send Feedback

What will be the output of the following code ?

```
public class Main {
   public static void change(int input[]){
   input = new int[5];
   input[0] = 15;
   }

   public static void main(String args[]){
      int arr[] = new int[5];
      change(arr);
      System.out.println(arr[0]);
   }
}
```

Options

Attempts left: 1/2

This problem has only one correct answer

- ○ 15
- ◉ 0
- ○ Error

Hurray! Correct Answer

Solution Description

####"arr" is a reference to the array, which contains address of the array. Hence we have passed this address to function "change". So input and arr will refer to same array initially, but in the first statement in function "change" input if made to refer to new integer array. So input[0] = 15, will change the first element of this newly formed array and the array created in main will remain unchanged.

Predict The Output

Send Feedback

What will be the output of the following code?

```
public static int[] change(int input[]){
input = new int[5];
input[0] = 15;
return input;
}

public static void main(String args[]){
   int arr[] = new int[5];
   arr=change(arr);
   System.out.println(arr[0]);
}
```

Options

Attempts left: 1/2

This problem has only one correct answer

- ○ 0
- ◉ 15
- ○ error

Hurray! Correct Answer

Return_Array_Sum

Send Feedback

Given an array/list(ARR) of length N, you need to find and return the sum of all the elements in the 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.

The 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 sum of the numbers in the array/list.

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

3

```
9 8 9
Sample Output 1:
26
Sample Input 2:
2
5
1 2 3 4 5
3
10 20 30
Sample Output 2:
15
60


public class Solution {

   public static int sum(int[] arr) {
       //Your code goes here
       int sum = 0;
       for(int i=0; i<arr.length; i++){
           sum += arr[i];
       }
       return sum;
   }
}
```

Linear Search

Send Feedback

You have been given a random integer array/list(ARR) of size N, and an integer X. You need to search for the integer *X* in the given array/list using 'Linear Search'.

You have been required to return the index at which *X* is present in the array/list. If *X* has multiple occurrences in the array/list, then you need to return the index at which the first occurrence of *X* would be encountered. In case *X* is not present in the array/list, then return -1.

'Linear search' is a method for finding an element within an array/list. It sequentially checks each element of the array/list until a match is found or the whole array/list has been searched.

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.

Third line contains the value of X(integer to be searched in the given array/list)

Output format :

For each test case, print the index at which *X* is present or -1, otherwise.

```
Output for every test case will be printed in a separate line.
Constraints :
1 <= t <= 10^2
0 <= N <= 10^5
-2 ^ 31 <= X <= (2 ^ 31) - 1
Time Limit: 1 sec
Sample Input 1:
1
7
2 13 4 1 3 6 28
3
Sample Output 1:
4
Sample Input 2:
2
7
2 13 4 1 3 6 28
9
5
7 8 5 9 5
5
Sample Output 2:
-1
2
```

```
public class Solution {

   public static int linearSearch(int arr[], int x) {
       //Your code goes here
       for(int i=0; i<arr.length; i++){
           if(arr[i]==x){
               return i;
           }
       }
       return -1;
   }
}
```

Arrange_Numbers_In_Array

Send Feedback

You have been given an empty array(ARR) and its size N. The only input taken from the user will be *N* and you need not worry about the array.

Your task is to populate the array using the integer values in the range 1 to N(both inclusive) in the order - 1,3,5,.....,6,4,2.

Note:

You need not print the array. You only need to populate it.

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.

The first and the only line of each test case or query contains an integer 'N'.
Output Format :
For each test case, print the elements of the array/list separated by a single space.

Output for every test case will be printed in a separate line.
Constraints :
1 <= t <= 10^2
0 <= N <= 10^4

Time Limit: 1sec
Sample Input 1 :
1
6
Sample Output 1 :
1 3 5 6 4 2
Explanation of Sample Input 1 :
Since the value of N is 6, the number will be stored in the array in such a fashion that 1 will appear at 0th index, then 2 at the last index, in a similar fashion 3 is stored at index 1. Hence the array becomes 1 3 5 6 4 2.
Sample Input 2 :
2
9
3
Sample Output 2 :
1 3 5 7 9 8 6 4 2
1 3 2

```
public class Solution {

   public static void arrange(int[] arr, int n) {
       //Your code goes here
       if(n%2==0){
           for(int i=0, j=1; i<n/2; j = j+2, i++){
               arr[i] = j;
           }
           for(int i=(n/2), j=(n/2*2); i<n; j = j-2, i++){
               arr[i] = j;
           }
       }
       else
       {
           for(int i=0, j=1; i<=n/2; j = j+2, i++){
               arr[i] = j;
           }
           for(int i=(n/2)+1, j=(n; i<n; j = j-2, i++){
               arr[i] = j;
           }
       }

   }
```

```
}
```

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
```

```
public class Solution {

   public static void swapAlternate(int arr[]) {
       //Your code goes here
       for(int i=0; i<arr.length-1; i = i+2){
           arr[i] = arr[i]+arr[i+1];
           arr[i+1] = arr[i]-arr[i+1];
           arr[i] = arr[i]-arr[i+1];
```

```
        }
    }
}
```

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
```

```
public class Solution {

   public static int findUnique(int[] arr) {
       // Your code goes here
       boolean[] flag = new boolean[arr.length];
       int res = 0;
       // for(int i=0; i<arr.length; i++){
```

```
        //      flag[i]=true;
        // }
        // for (int i = 0; i < arr.length-1; i++) {
        //      for (int j = i+1; j < arr.length; j++) {
        //          if (flag[i]==false && arr[i] == arr[j]) {
        //              flag[i]=flag[j]=true;
        //              break;
        //          }
        //      }
        // }
        // for(int i=0; i<arr.length; i++)
        //      if (flag[i]==false)
        //          return arr[i];
        for(int i=0; i<arr.length;i++)
            res ^= arr[i];
        return res;
    }
}
```

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

public class Solution {

   public static int duplicateNumber(int arr[]) {
       //Your code goes here
       // for(int i=0; i<arr.length; i++){
       // for(int j=0; j<arr.length; j++){
       // if(i==j)
       // continue;
       // if(arr[i]==arr[j])
       // return arr[i];
       // }
       // }
       // return -1;
       int ans = 0;
       for(int i=0; i<arr.length; i++){
           ans ^= arr[i];
       }
       for(int i=0; i<=arr.length-2; i++){
           ans ^= i;
       }
       return ans;
   }
}
```

Intersection of Two Arrays II

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 the 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 <= 10^3

0 <= M <= 10^3

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 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.

```
public class Solution{

   public static void intersections(int arr1[], int arr2[]) {
       //Your code goes here

       for(int i=0; i<arr1.length; i++){
           for(int j=0; j<arr2.length; j++){
```

```
                if(arr1[i]==arr2[j]){
                    System.out.print(arr2[j]+" ");
                    arr2[j]=-1;
                    break;
                }
            }
        }
        System.out.println();
    }
}
```

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 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).


public class Solution {

   public static int pairSum(int arr[], int x) {
       //Your code goes here
       int count = 0;
       for(int i=0; i<arr.length; i++){
           for(int j=i+1; j<arr.length; j++){
               if(i!=j && arr[i]+arr[j]==x)
                   count++;
           }
       }
       return count;
   }
}
```

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)


public class Solution {

   public static int findTriplet(int[] arr, int x) {
       //Your code goes here
    int count = 0;
       for(int i=0; i<arr.length; i++){
           for(int j=i+1; j<arr.length; j++){
               for(int k=j+1; k<arr.length; k++){
               if(arr[i]+arr[j]+arr[k]==x)
                   count++;
               }
           }
       }
       return count;
   }

}
```

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 in 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

```
public class Solution {

   public static void swap(int arr[], int a, int b){
       arr[a] = arr[a]+arr[b];
       arr[b] = arr[a]-arr[b];
       arr[a] = arr[a]-arr[b];

       }

   public static void sortZeroesAndOne(int[] arr) {
       //Your code goes here
       int marker = 0;
       for(int i=1; i<arr.length; i++){
           if(arr[marker]==arr[i])
               continue;
```

```
            else if(arr[marker]>arr[i]){
                swap(arr,marker,i);
                marker++;
            }
            else
                marker = i;

        }
    }
}
```