

First Index And Last Index (1 july)

Problem

Submissions

Leaderboard

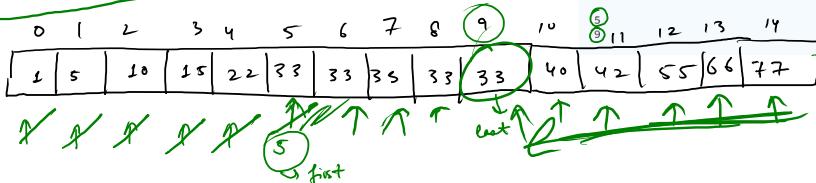
Discussions

1. You are given a number n, representing the size of array a.
2. You are given n numbers, representing elements of array a.
- Assumption - Array is sorted. Array may have duplicate values.

3. d →

$n = 15$

$d = 33$



left to right (9) → last occurrence of 33

right to left → first occurrence

Sample Input 0

15
1
5
10
15
22
33
33
33
33
33
40
42
55
66
77

Sample Output 0

$d \rightarrow$

First index

```
for (int i=0; i<n; i++)  
{  
    if (arr[i] == d)  
        return i;  
}  
return -1;
```

Last Index

↳ for (int i=n-1; i>=0; i--)
{
 if (arr[i] == d)
 return i;
}

↳

return -1;

$\text{ans} = -1$

```
for (int i=0; i<n; i++)  
{  
    if (arr[i] == d)  
        ans = i;  
}
```

$\text{ans} = 9$

$\text{ans} = 9$?
return ans;

$d = 2$

 ans = -1 $i = 0 \leftarrow \text{f}(T)$
 1 = -2 X $i = -2 \leftarrow \text{F}$
 $i = 0 \leftarrow \text{f}(T)$
 1 = -2 X $i = 1 \leftarrow \text{f}(T)$
 2 = -2 $i = 2 \leftarrow \text{f}(T)$
 $i = 1 \leftarrow \text{f}(T)$
 2 = -2 $i = 2 \leftarrow \text{f}(T)$
 ans = 1
 $i = 2 \leftarrow \text{f}(T)$
 2 = -2 $i = 2 \leftarrow \text{f}(T)$
 ans = 2

```

public class Solution {
    public static int firstIndex(int arr[], int d) {
        for (int i=0; i<arr.length; i++) {
            if (arr[i] == d) {
                return i;
            }
        }
        return -1;
    }

    public static int lastIndex(int arr[], int d) {
        for (int i=arr.length-1; i>=0; i--) {
            if (arr[i] == d) {
                return i;
            }
        }
        return -1;
    }

    public static int lastIndex2(int arr[], int d) {
        int ans = -1;
        for (int i=0; i<arr.length; i++) {
            if (arr[i] == d) {
                ans = i;
            }
        }
        return ans;
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];
        for (int i=0; i<n; i++) {
            arr[i] = scn.nextInt();
        }

        int d = scn.nextInt();

        System.out.println(firstIndex(arr, d));
        System.out.println(lastIndex2(arr, d));
    }
}
  
```

$i = 3 \leftarrow \text{f}(T)$
 $2 = -2 \leftarrow \text{f}(T)$
 ans = 3

$i = 4 \leftarrow \text{f}(T)$
 $2 = -2 \leftarrow \text{f}(T)$
 ans = 4 → lastIndex of 2

$i = 5 \leftarrow \text{f}(T)$
 $3 = -2 \leftarrow \text{f}(T)$
 $i = 6 \leftarrow \text{f}(T)$
 $6 = -2 \leftarrow \text{f}(T)$
 $i = 7 \leftarrow \text{f}(F)$

*Stop where find the first occurrence of 1+
 all the elements of array*

1. You are given an array of size 'n' and n elements of the same array.
2. You are required to find and print all the subarrays of the given array.
3. Each subarray should be space separated and on a separate lines. Refer to sample input and output.

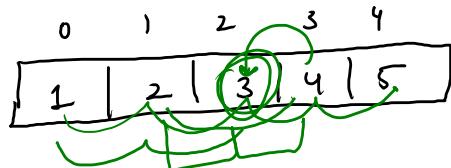
Sample Input 0

5 → n
1
2
3
4
5

$$n = 5$$

1 | 2 | 3 | 4 | 5

Sub array \Rightarrow continuous part of an array



a 1 2 3 4 5 ✓

1 2 3 → ✓

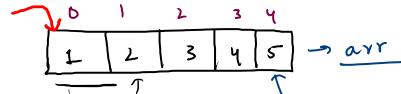
b 2 4 5 ✗

1 3 5 → ✗

c 2 4 3 ✗

d 5 3 1 ✗

e 2 3 4 ✓



How many subarrays are possible for 'n' elements?

Starting from 1	Starting from 2
1	2
1 2	2 3
1 2 3	2 3 4
1 2 3 4	2 3 4 5
1 2 3 4 5	

Starting from 3	Starting from 4	Starting from 5
3	4	5
3 4	4 5	
3 4 5		

→ Dry Run
+
→ Code

$$\text{Total no.} = 5 + 4 + 3 + 2 + 1 \\ = 15$$

Total no. of subarrays for 'n' elements =

$$n + (n-1) + (n-2) + (n-3) + \dots - 1$$

↳ Sum of all natural nos.

Total no. of subarrays. = $\frac{n(n+1)}{2}$

```

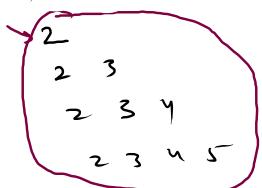
public static void subArray(int arr[]){
    int n = arr.length;
    for(int i = 0; i < n; i++){ // stating Index
        for(int j = i; j < n; j++){ // Choosing ending index
            for(int k = i; k < j; k++){ // For Printing
                System.out.print(arr[k] + " ");
            }
            System.out.println();
        }
    }
}

```

} 1 2 3
 → 1 2 2 3
 → 1 2 3
 →

0	1	2
1	2	3

→ \downarrow starting | ending
 1
 1 2
 1 2 3
 1 2 3 4
 1 2 3 u 5



$i = 0 \leq 3$ (T)
 $j = 0 \leq 3$ (T)
 $k = 0 \leq 0$ (T)
 $k = 1 \leq 0$ (F)
 $j = 1 \leq 3$ (T)
 $k = 0 \leq 1$ (T)
 $k = 1 \leq 1$ (T)
 $k = 2 \leq 1$ (F)
 $j = 2 \leq 3$ (T)

3 4 5
 3 4
 3 4 5
 — —

why j loops with
 not starting with
 0

$i = 1 \leq 3$ (T)
 $j = 0 \leq 3$ (T)
 $k = 1 \leq 0$ (F)

Hint:-

- ↳ 1 Taking an element as the starting pt.
- ↳ 2 Ending index
- ↳ 3 Printing all the elements.

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void subArray(int arr[]){
        int n =arr.length;
        for(int i =0;i<n;i++){// stating Index
            for(int j=i;j<n;j++){ // Choosing ending index
                for(int k=i;k<=j;k++){ // For Printing
                    System.out.print(arr[k] + " ");
                }
                System.out.println();
            }
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n =scn.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        subArray(arr);
    }
}
```

Print Alternate Array Elements Linewise (1 july)

Problem

Submissions

Leaderboard

Discussions

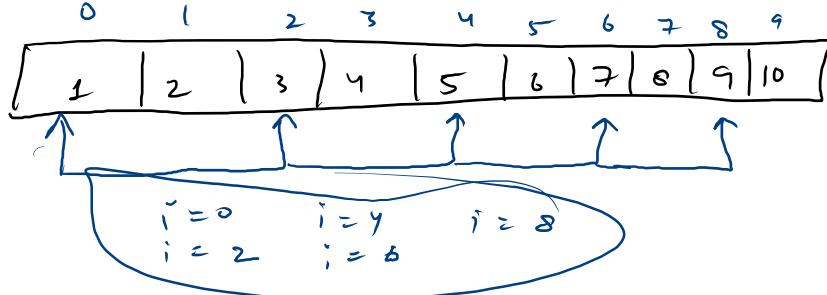
Sample Input 0

```
10
1
2
3
4
5
6
7
8
9
10
```

Take n as an integer input. Declare an array of size n that stores value of int data-type. Then take n integer inputs and store them in the array one by one. Then print the alternate elements of the array starting from the 0th index

Sample Output 0

```
1
3
5
7
9
```



```
for (int i=0; i<n; i+=2)
```

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr [] = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        for(int i=0;i<n;i+=2){
            System.out.println(arr[i]);
        }
    }
}
```

Count Odd Pair (2 July)

Problem

Submissions

Leaderboard

Discussions

Take the array of size n and their values from user. And Find Pairs whose sum is odd.

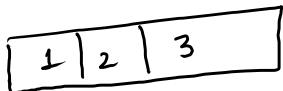
Sample Input 0

3
1 2 3

Sample Output 0

1 2
2 1
2 3
3 2

$n = 3$



sum pair

odd

pair pair

(2, 3)

(3, 1), (3, 2), (3, 3)

(1, 2), (1, 3),
(2, 1), (2, 3)

```

public class Solution {
    public static void printOddPairs(int arr[]){
        for(int i=0; i<arr.length; i++){
            for(int j=0; j<arr.length; j++){
                int sum = arr[i]+arr[j];
                if(sum%2 != 0){
                    System.out.println(arr[i] + " " + arr[j]);
                }
            }
        }
    }
}

```

$$TC \rightarrow O(n^2)$$

$$SC = O(1)$$



$$\begin{array}{l}
 i=0 \quad j \rightarrow n \\
 i=1 \quad j \rightarrow n \\
 i=n \quad j \rightarrow n
 \end{array}$$

$$n+n+n - n \text{ times}$$

$$n \times n - n^2$$

\hookrightarrow
 \approx

$$\begin{matrix}
 & 1 & 2 \\
 1 & & \\
 2 & 1
 \end{matrix}$$

$$i = 0 < 3 \text{ (T)}$$

$$j = 0 < 3 \text{ (T)}$$

$$\text{sum} = arr[0] + arr[0] = 1+1 = 2$$

$$i = 1 < 3 \text{ (T)}$$

$$\text{sum} = arr[0] + arr[1] = 1+2 = 3$$

$$i = 2 < 3 \text{ (T)}$$

$$\text{sum} = arr[1] + arr[2] = 1+3 = 4 \times$$

$$i = 3 < 3 \text{ (F)}$$

$$i = 1 < 3 \text{ (T)}$$

$$j = 0 < 3 \text{ (T)}$$

$$\text{sum} = arr[1] + arr[0] = 2+1 = 3$$

$$i = 1 < 3 \text{ (T)}$$

$$\text{sum} = arr[1] + arr[1] = 2+2 = 4 \times$$