

Field Return Part Number swwsww Summary

Overall Information

Registration Number: swwsw

Date: Fri 17 5:30

Time: 5:30

OECD: swwsw

Time: 5:30

Parts Recieved: swwsw

Courier/Docker Detail: swwsw

Complaint reported at BDS: swwsw

WJC No.: swwsw

FIP Type/Part No.: swwsw

Failure Hours: swwsw

Location BDS: swwsw

Investigation Details: swwsw

Date of closing: swwsw

Final State: swwsw

Pre-Investigation details

Date: Fri 17 5:30

Pump Part Number: 5:30

Serial Number (with MFD): swwsw

LAC: 5:30

eJC Number: swwsw

Dealer : swwsw

Application: swwsw

Voice of Customer: swwsw

Failure Hours: swwsw

Seal condition: swwsw

Observations

Date: Fri 17 5:30

Fuel Related Issues: swwsw

Hitting mark on timer plate: 5:30

Blueish mark on camplate/rollers: swwsw

Pitting on camplate (Inner edge/outer edge): swwsw

Number of camplate lobes damaged: swwsw

Number of roller damaged: swwsw

Rubbing marks - cross disk: swwsw

Delta: swwsw

Rubbing mark - drive shaft claw: swwsw

Images

```
Code Ends
{
// Driver Code Starts
// Input array
// to find the sum of contiguous subarray with maximum sum.
maxSubarraySum(int arr[], int n){
    maxEndingHere = arr[0];
    maxSoFar = arr[0];
    for(int i=1; i<n; i++){
        maxEndingHere = max( arr[i] , maxEndingHere+arr[i]);
        maxSoFar = max(maxSoFar , maxEndingHere);
    }
    return maxSoFar;
}
Code Ends
```

```
int large = arr[0], large2 = -1;
for(int i=1; i<n; i++)
{
    if(arr[i] > large)
    {
        large2 = large;
        large = arr[i];
    }
    else if(arr[i] < large)
    {
        large2 = large2 > arr[i] ? large2 : arr[i];
    }
}
return large2;
```

Given an array **Arr** of size **N**, print second largest element from an array.

Example 1:

Input:

N = 6

Arr[] = {12, 35, 1, 10, 34, 1}

Output: 34

Explanation: The largest element of the array is 35 and the second largest element is 34.

Example 2:

Input:

N = 3

Arr[] = {10, 5, 10}

Practice

Editorial Submissions Doubt Support

14% Submissions: 43317 Points: 0

Given an array **Arr** of size **N**, print second largest element from an array.

Input: N = 6, Arr[] = {12, 35, 1, 10, 34, 1}

Output: 34

Explanation: The largest element of the array is 35 and the second largest element is 34.

```
1 // Driver Code Starts
2 #include <bits/stdc++.h>
3 using namespace std;
4 // Driver Code Ends
5 // User function template for C++
6 class Solution {
7 public:
8     // Function returns the second
9     // largest element
10    int printSecondLargest(int arr[], int n) {
11        int largest1=0;
12        int largest2=0;
13        for(int i=0; i<n; i++)
14        {
15            if(arr[i]>largest1)
16            {
17                largest2=largest1;
18                largest1=arr[i];
19            }
20            else if(arr[i]>largest2 && arr[i]!=largest1)
21            {
22                largest2=arr[i];
23            }
24        }
25        return largest2;
26    }
27 };
28 // Driver Code Starts
29 int main() {
30     int t;
31     cin >> t;
32     while(t--){
33         int n;
34         cin >> n;
35         int arr[n];
36         for(int i = 0; i < n; i++)
37             cin >> arr[i];
38         Solution obj;
39         int ans=obj.printSecondLargest(arr, n);
40         cout<<ans<<endl;
41     }
42 }
```

Compile

Kadane's Algorithm

$$\text{maxEndingHere} = \max \left\{ \begin{array}{l} \text{maxEndingHere} + \text{num} \\ \text{num} \end{array} \right.$$

$$\text{maxSoFar} = \max \left\{ \begin{array}{l} \text{maxSoFar} \\ \text{maxEndingHere} \end{array} \right.$$

[3, 5, -9, 1, 3, -2, 3, 4, 7, 2, -9, 6, 3, 1] -5, 3, 8, -1, 1, 4, 2, 5, 9, 16, 18, 9, 15, 18, 13, 14, 3, 8, 8, 8, 8, 8, 9, 16, 18, 18, 18, 13, 13

$$\text{Time} = O(N)$$

$$\text{Space} = O(1)$$

Express, MongoDB & More: The Complete Bootcamp 2022

Notes Announcements Reviews Learning tools

process.env

Results for "process" (78 lectures)