

n - - - - - }
 (✓ - - - - -) }
 between
 - - - - -
 }
 }

<p>✓ foo (- - - - -) {</p> <p>1st line code</p> <p>2nd " "</p> <p>3rd " "</p> <p>· · · · ·</p> <p>}</p>	<p>foo (- - - - -) {</p> <p>1st line</p> <p>2nd line</p> <p>}</p>
--	---

foo (- - - - -) {
 { if (- - - - -)
 sys }
 }

1st line

2nd line

Declare the **first array** of size **n** that stores values of **int** data-type. Then take **n** integer inputs and store them in the array one by one.

Print **Product** of all the elements of arrays

Input Format

First line contains **N** as Size of Array.

Second line contains **n** integer value as **Arr[i]** values

Constraints

NA

Output Format

Product of all the array values

Sample Input 0

```
5
1
2
3
4
5
```

Sample Output 0

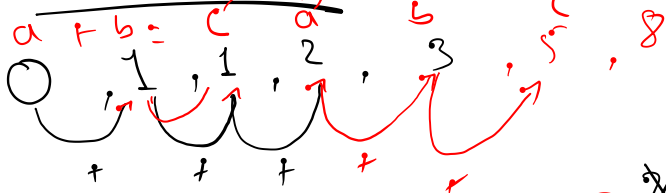
```
120
```

Explanation 0

1*2*3*4*5

5
1 2 3 4 5
5

fibonacci



0 1 1 new

a = 0
b = 1
c = a + b = 1

{ 0, 1, 1, 2, 3, 5, ... }

a = 0
b = 1

0, 1, 1, 2

for (int i = 2; i < n; i++) {

c = a + b
a = b
b = c

10
a = 0
b = 1
c = a + b

You have been given a **random integer array/list (ARR)** of size **N**. You are required to find and return the **second largest** element present in the array/list.

If **N <= 1** or all the elements are same in the array/list then return **-2147483648** or **-2³¹** (It is the smallest value for the range of Integer)

Input Format

The first line contains an integer '**N**' representing the size of the array/list.

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

Constraints

0 <= N <= 10⁵

Output Format

Print the **second largest** in the array/list if exists, **-2147483648** otherwise.

Output will be printed in a separate line.

Sample Input 0

7
2 13 4 1 3 6 28

Sample Output 0

13

Sample Input 1

Submitted Code

```
Language: java 15
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         int arr[] = new int[n];
10        for(int i=0; i<n; i++) arr[i] = sc.nextInt();
11        System.out.print(secondMax(arr));
12    }
13    public static int secondMax(int arr[]){
14        int fMax=Integer.MIN_VALUE;
15        int sMax=Integer.MIN_VALUE;
16
17        for(int i=0; i<arr.length; i++){
18            if(arr[i]>fMax){
19                sMax=fMax;
20                fMax=arr[i];
21            }else if(arr[i]<fMax && arr[i]>sMax){
22                sMax=arr[i];
23            }
24        }
25        return sMax;
26    }
27 }
```

2nd highest
(2, 13, 4, 1, 6, 2, 28)
28
13

first Max = Integer.MIN
second Max = Integer.MIN

for(int i=0; i<n; i++){
if(arr[i]>fMax){
sMax = fMax;
fMax = arr[i];

else if (arr[i]<fMax && arr[i]>sMax){
sMax = arr[i];

28
sMax = fMax
(20, 30 | 50, 40, 8)
fMax = 50
sMax = 40

