

STORING MARKS OF 5 STUDENTS

We are given 5 numbers in the input, which represent the marks of 5 students. We just need to store them in our program.

```
int main()
{
    int m1, m2, m3, m4, m5;
    cin >> m1 >> m2 >> m3 >> m4 >> m5;
```

STORING MARKS OF 10 STUDENTS

We are given 10 numbers in the input, which represent the marks of 10 students. We just need to store them in our program.

```
int main()
{
    int m1, m2, m3, m4, m5, m6, m7, m8, m9, m10;
    cin >> m1 >> m2 >> m3 >> m4 >> m5 >> m6 >> m7 >> m8 >> m9 >> m10;
```

STORING MARKS OF 100 STUDENTS

We are given 100 numbers in the input, which represent the marks of 100 students. We just need to store them in our program.

```
// impossible and very hard to manage
int main()
{
    int m1, m2, m3, ... m100;
    cin >> m1 >> m2 >> m3 >> ... >> m100;
}
```



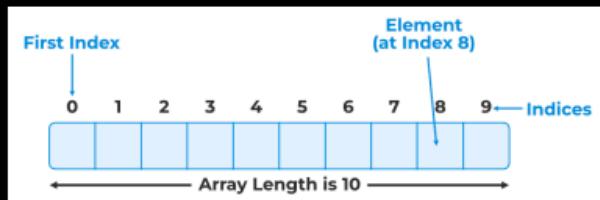
STORING MARKS OF 100 STUDENTS

We are given 100 numbers in the input, which represent the marks of 100 students. We just need to store them in our program.

```
int main () {  
    marks [ ] = { 17, 25, ... , 75 };  
    int marks [100];  
    marks [0] = 17;  
    marks [4] = 25;  
    marks [99] = 75;  
  
    cin >> marks [0];  
    cin >> marks [1];  
    cin >> marks [2];  
    ...  
    cin >> marks [99];  
}  
for ( i = 0; i <= 99; i++ )  
{  
    cin >> marks [i];  
}
```

ARRAYS

- A **collection** of **similar kind of data** stored together at **continuous memory locations**.
 - It helps us **create multiple variables** of the **same data type** using a **single name**.
 - Each element can be accessed using the **array name** along with its **unique index**.
 - Indexing starts from **0**, and the last element is located at index **N - 1**, where N is the total number of elements in the array.



CREATING AN ARRAY

- *DataType* **ArrayName[NUMBER_OF_ELEMENTS];**

◦ int a[20];

• **float b[30];**

$b =$	$\boxed{12.4}$	$\boxed{12.7}$	$\boxed{1.5}$	$\boxed{1}$	$\boxed{1}$	\dots	$\boxed{1}$
	$\underline{0}$	$\underline{1}$	$\underline{2}$	$\underline{3}$	\dots	\dots	$\underline{9}$

long long c[500];

c: [] - - - - - - - -]

- Once you create an array, all elements can have any random value (often referred as **garbage value**)

ACCESSING ELEMENTS

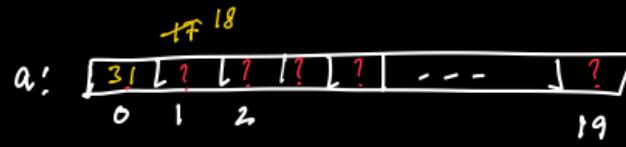
(0 to N-1)

- int a[20];

- a[0] = 31;

- a[1]++;

- cout << a[2] << endl; // Garbage Value



a[1] ✓

a[19] ✓

a[20] ✗

a[-1] ✗

- Array indices are from 0 to (N-1), if you try to access a[i] where i is not a valid index (i.e. i < 0 or i >= N), it will lead to undefined behaviour, could be a runtime error, segmentation fault or accessing any garbage value.

Example

LEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

urce:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     // 0. 1. 2. 3 .. 99
6     int a[100]; // [ , , , , , , , , , ]
7
8     cout << a[2] << endl;
9
10    cout << a[5] << endl;
11
12    cout << a[3] << endl;
13
14    cout << a[99] << endl;
15
16 }
```

Switch off editor

Tab size: 4

Run

Language:



Input:

Dhruv

Choose file No file chosen
No more than 256 KB

Output:

```
1
548274176
4335669
4238555
```

=====

```
Used: 31 ms, 20 KB
```

First 255 bytes only

INPUT / OUTPUT

5
78124

78124

```
int main()
{
    int n;
    cin >> n;

    int A[n];
    for(int i = 0; i < n; i++)
    {
        cin >> A[i];
    }

    for(int i = 0; i < n; i++)
    {
        cout << A[i] << " ";
    }
}
```

Online C++ Compiler and Visualizer

C++ Run Visualize Code

Learning C++? Check out our com

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     int a[n];
8
9     // indexing: 0 to n - 1
10    for(int i = 0; i <= n - 1; i++) {
11        cin >> a[i];
12    }
13    for(int i = 0; i <= n - 1; i++) {
14        cout << a[i] << " ";
15    }
16}
17
18
19
20
21 }
```

Dhruv

Output

Status : Successfully executed

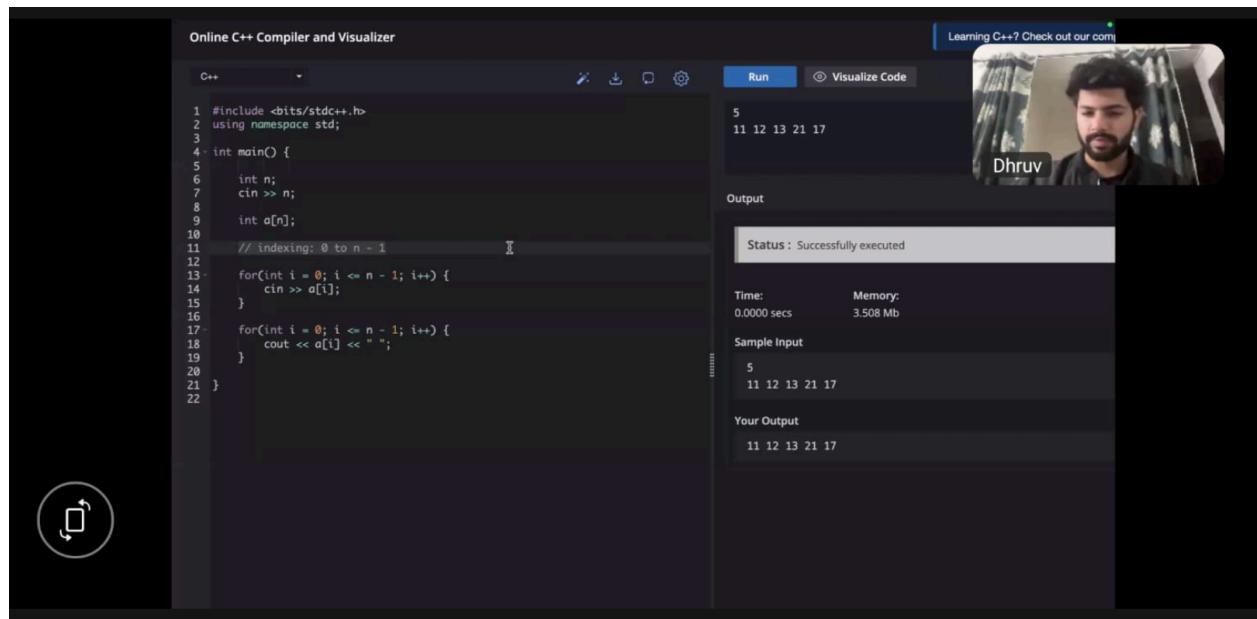
Time: 0.0000 secs Memory: 3.508 Mb

Sample Input

5
11 12 13 21 17

Your Output

11 12 13 21 17



PRINT ARRAY IN REVERSE

5
7 8 1 2 4

4 2 1 8 7

 **CODEFORCES**
Sponsored by TON

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7
8     int a[n];
9     for(int i = 0; i <= n - 1; i++) {
10        cin >> a[i];
11    }
12
13
14
15    for(int i = n - 1; i >= 0; i--) {
16        cout << a[i] << " ";
17    }
18
19 }
```

Switch off editor Tab size: 4

Run

Language:

Input:
7
11 12 13 21 17 11 8

Choose file No file chosen
No more than 256 KB

Output:
8 11 17 21 13 12
=====

Used: 15 ms, 0 KB

First 255 bytes only

Codeforces (c) Copyright 2010-2026 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Feb/05/2026 21:15:27 UTC+5.5 (I1).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#) | [Terms and Conditions](#)



Problem: Sum of Array

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

B. Sum of Array

time limit per test: 1 second
memory limit per test: 256 megabytes

You are given an array of integers.

Find the sum of all elements of the array.

Input

- First line contains integer N ($1 \leq N \leq 10^5$) — size of the array.
- Second line contains N integers A_1, A_2, \dots, A_N ($-10^9 \leq A_i \leq 10^9$).

Output

Print a single integer — the sum of the array elements.

Example

input	Copy
5 1 2 3 4 5	
output	Copy
15	



Dhruv

About Group

100 school

Welcome to the official group of Bootcamp 1.0. Here, you will find assignment and practice problems.

Group website

The example input is available in the clipboard.

→ Group

• Arrays

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5
6     int n;
7     cin >> n;
8
9     int a[n];
10    for(int i = 0; i <= n - 1; i++) {
11        cin >> a[i];
12    }
13
14    long long sum = 0;
15    for(int i = 0; i <= n - 1; i++) {
16        sum += a[i];
17    }
18
19    cout << sum << endl;
20}
21
```

Switch off editor

Tab size: 4

Run



Dhruv

Language: C++

Input: 5
1 2 3 4 5

Choose file | No file chosen
No more than 256 KB

Output: Running...

First 255 bytes only

CodeForces (c) Copyright 2010-2026 Mike Mirzayanov

FIND MAX OF THE ARRAY

6
148793

6
148793

g

g

The screenshot shows a video call interface. On the left, there is a presentation slide with two blue boxes containing the number 6 and the array 148793. Below the slide is a code editor window displaying C++ code for finding the maximum element in an array. On the right, there is a video feed of a person named Dhruv.

Target for RCB: 204 Can Smriti lead RCB to ...

```
int ans = a[0];  
  
for(int i = 1; i < n; i++)  
{  
    if(a[i] > ans)  
    {  
        ans = a[i];  
    }  
}  
  
cout << ans << endl;
```

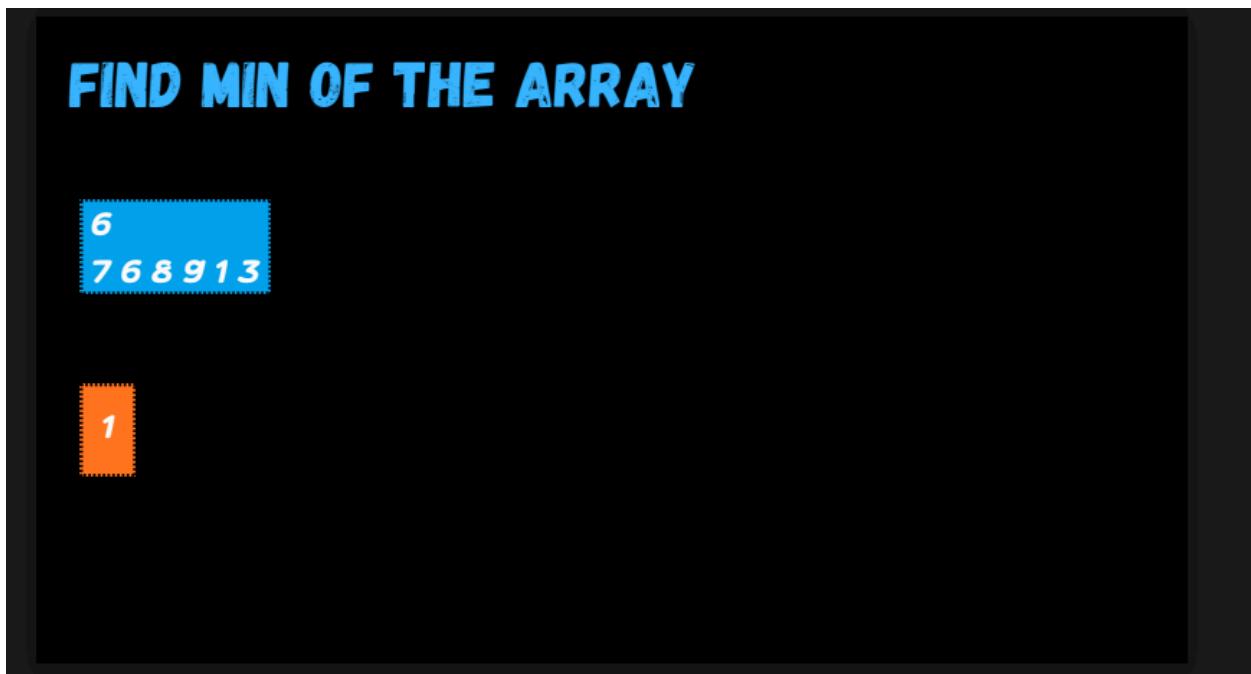
Find max and location of the element in array

The screenshot shows a video call interface. On the left, there is a digital whiteboard or canvas with a grid background. Inside the grid, the number '6' is highlighted in a blue box at the top-left. Below it, the numbers '1 4 8 7 9 3' are listed. To the right of the grid, there is handwritten text: 'Max element' at the top, followed by 'ans' with a circled '1' above it, and 'location' with a circled '1' below it. Below this, the numbers '9 8 4' are written vertically, with '9' having a circled '1' above it. At the bottom of the grid, there is handwritten text: '3' with a circled '1' above it, and '5'. On the right side of the screen, a video feed of a person with dark hair and a beard is visible, labeled 'Dhruv'. Below the video feed, there is some small text and icons.

```
int ans = a[0], location = 1;

for(int i = 1; i < n; i++)
{
    if(a[i] > ans)
    {
        ans = a[i];
        location = i + 1;
    }
}

cout << ans << " " << location << endl;
```



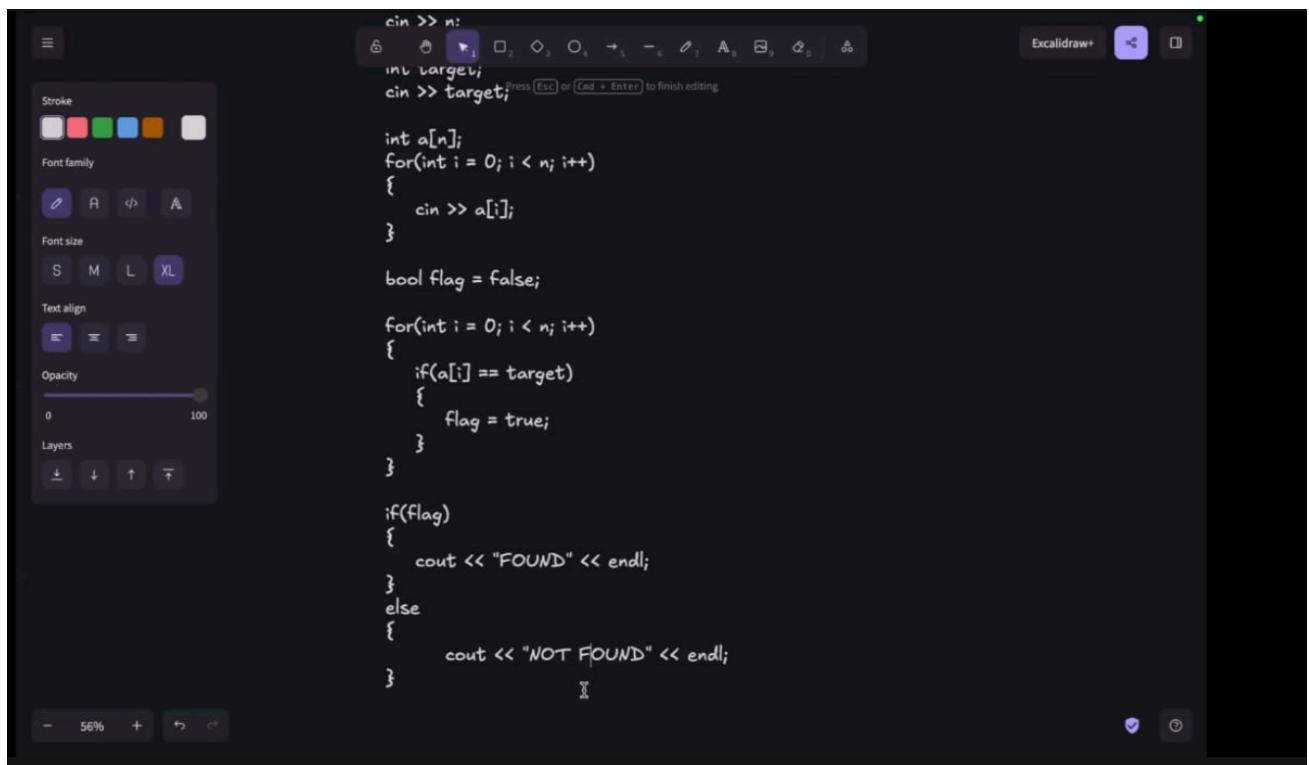
SEARCH IN AN ARRAY

6 10
7 6 8 9 13

6 8
7 6 8 9 13

NOT FOUND

FOUND



The screenshot shows a dark-themed interface for a code editor, likely within the Excalidraw+ application. On the left, there is a sidebar with various tools and settings: 'Stroke' (color palette), 'Font family' (dropdown), 'Font size' (S, M, L, XL), 'Text align' (center), 'Opacity' (0-100 slider), and 'Layers' (button). The main area contains the following C++ code:

```
cin >> n;
int target;
cin >> target;

int a[n];
for(int i = 0; i < n; i++)
{
    cin >> a[i];
}

bool flag = false;
for(int i = 0; i < n; i++)
{
    if(a[i] == target)
    {
        flag = true;
    }
}

if(flag)
{
    cout << "FOUND" << endl;
}
else
{
    cout << "NOT FOUND" << endl;
}
```

The code reads an integer `n`, a target value, and `n` integers into an array `a`. It then iterates through the array to find the target value. If found, it prints "FOUND"; otherwise, it prints "NOT FOUND".

COUNT OCCURENCES

67
76 79 73

68
76 89 13

3

1

```
int n;
cin >> n;

int target;
cin >> target;

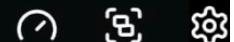
int a[n];
for(int i = 0; i < n; i++)
{
    cin >> a[i];
}

bool flag = false;
for(int i = 0; i < n; i++)
{
    if(a[i] == target)
    {
        count++;
    }
}

cout << count << endl;
```

7979721719

|> ▶ 01:14:49/02:31:05



CHECK IF THE GIVEN ARRAY IS SORTED

6
1 2 2 8 11 31

6
7 6 7 9 7 3

YES

NO

bool

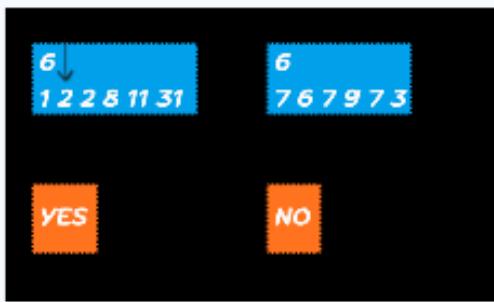
```
for(int i = 1; i < n; i++)
{
    // they are in correct order
    if(a[i] >= a[i - 1])
    {
        // no issues here
    }
    else
    {
        // array is not sorted
        flag = true;
        break;
    }
}

if(flag)
{
    cout << "NOT SORTED";
}
```

$a[i] < a[i - 1]$

flag

False



For an array to be in ascending order,

$a[i] \geq a[i - 1]$ this must be true for every i .

```
int n;
cin >> n;

int a[n];
for(int i = 0; i < n; i++)
{
    cin >> a[i];
}

bool flag = false;
for(int i = 1; i < n; i++)
{
    if(a[i] < a[i - 1])
    {
        // array is not sorted
        flag = true;
        break;
    }
}
if(flag)
{
    cout << "NOT SORTED";
}
else
{
    cout << "SORTED";
}
```

10 12 16 17 19

$a[i] < a[i - 1]$

false

flag

SORT 01

6
110110

6
100110

001111

000111

6
110110

6
100110

001111

000111

2 4
c0 c1

0000000.....111111111

We can only have 0's and 1's

Print this array in ascending order.

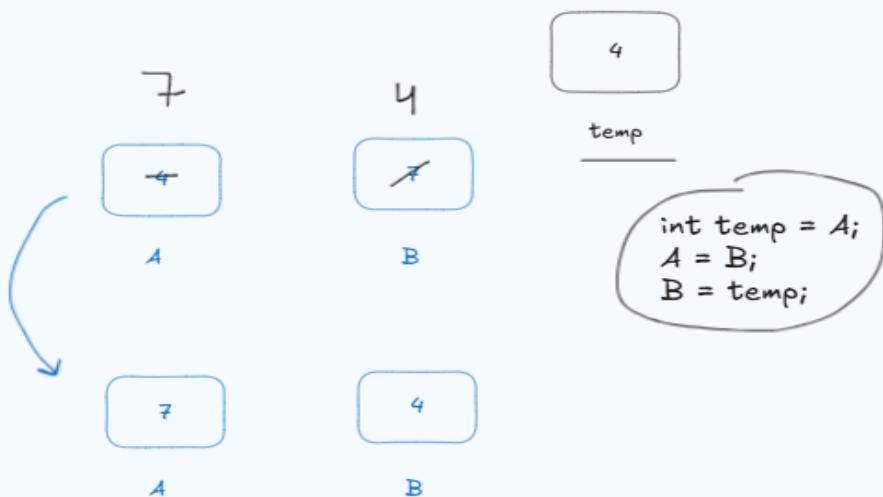
```
int c0 = 0, c1 = 0;  
  
for(int i = 0; i < n; i++)  
{  
    if(a[i] == 0)  
    {  
        c0++;  
    }  
    else  
    {  
        c1++;  
    }  
}  
  
// print all 0's  
for(int i = 0; i <= c0; i++)  
{  
    cout << "0 ";  
} _____  
  
// print all 1's  
for(int i = 0; i <= c1; i++)  
{  
    cout << "1 ";  
}
```

SWAP 2 NUMBERS

A = 2
B = 5

A = 5
B = 2

SWAP A and B



PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5
6     int a = 4;
7     int b = 7;
8
9     swap(a, b); I
10
11    cout << a << endl; // 7
12    cout << b << endl; // 4
13
14 }
```

79979721719

Language: GNU G-

Input:

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

Choose file No file chosen
No more than 256 KB

Output:

```
7
4
=====
Used: 31 ms, 0 KB
```

switch off editor 01:37:50/02:31:05 Tab size: 4

Run

First 255 bytes only

SWAP ALTERNATE

5
7 8 1 2 4

6
7 9 8 1 2 4

8 7 2 1 4

9 7 1 8 4 2

Swap alternate elements

5
7 8 1 2 4

6
7 9 8 1 2 4



8 7 2 1 4

9 7 1 8 4 2

```
for(int i = 1; i < n; i += 2)
{
    swap(a[i], a[i - 1]);
}
```

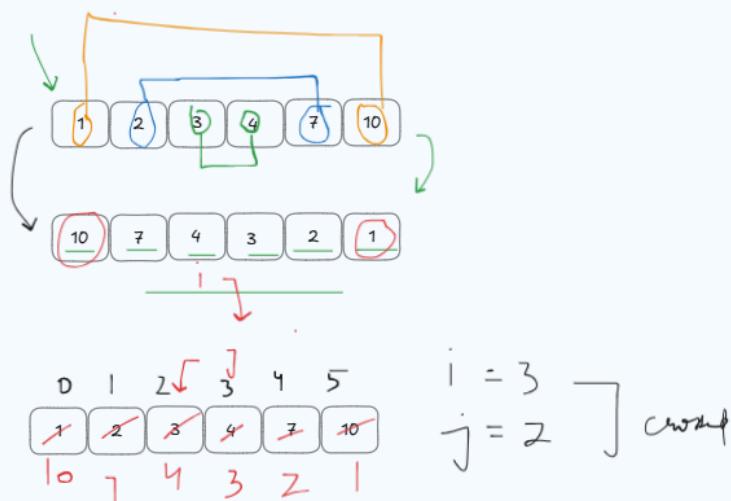
REVERSE IN THE SAME ARRAY

5
7 8 1 2 4

6
7 9 8 1 2 4

4 2 1 8 7

4 2 1 8 9 7



```
int i = 0, j = n - 1;
```

```
while(i <= j)
{
    swap(a[i], a[j]);
    i++;
    j--;
}
```

```
for(int i = 0; i < n; i++)
{
    cout << a[i] << " ";
```

MISSING NUMBER

Every number in the array is present twice, except for one special element that occurs only once. Find the special element.

7
2 1 9 1 2 3 9

7
1 2 1 2 3 4 3

3

4

MISSING NUMBER

Every number in the array is present twice, except for one special element that occurs only once. Find the special element.

7
2 1 9 1 2 3 9

7
1 2 1 2 3 4 3

3

4

```
int ans;  
for(int i = 0; i < n; i++)  
{  
    // test whether a[i] is the ans or not  
  
    int target = a[i];  
    int count = 0;  
  
    for(int j = 0; j < n; j++)  
    {  
        if(a[j] == target)  
        {  
            count++;  
        }  
    }  
  
    if(count == 1)  
    {  
        ans = a[i];  
        break;  
    }  
}  
  
cout << ans << endl;
```

Learning:

1. In cpp site some time they give number of test cases as input

Output
Print the expected array.

Examples

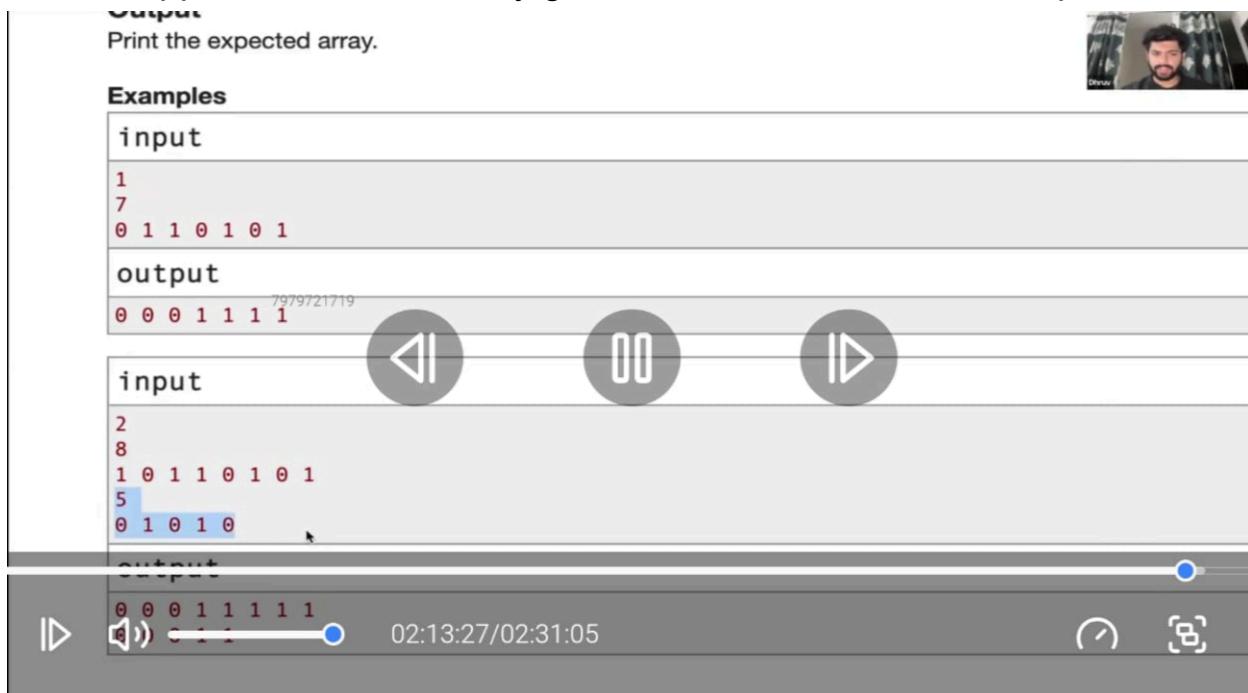
input
1
7
0 1 1 0 1 0 1

output
0 0 0 1 1 1 1 7979721719

input
2
8
1 0 1 1 0 1 0 1
5
0 1 0 1 0

output
0 0 0 1 1 1 1 1

02:13:27/02:31:05



HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT CUSTOM INVOCATION

Source:

```
30-    {  
31     cout << "0 ";  
32 }  
33  
34  
35 // print all 1's  
36 for(int i = 1; i <= c1; i++)  
37- {  
38     cout << "0 ";  
39 }  
40 }  
41  
42 int main()  
43 {  
44     int t;  
45     cin >> t;  
46  
47     for(int i = 1; i <= t; i++)  
48     {  
49         solve();  
50     }  
51 }
```

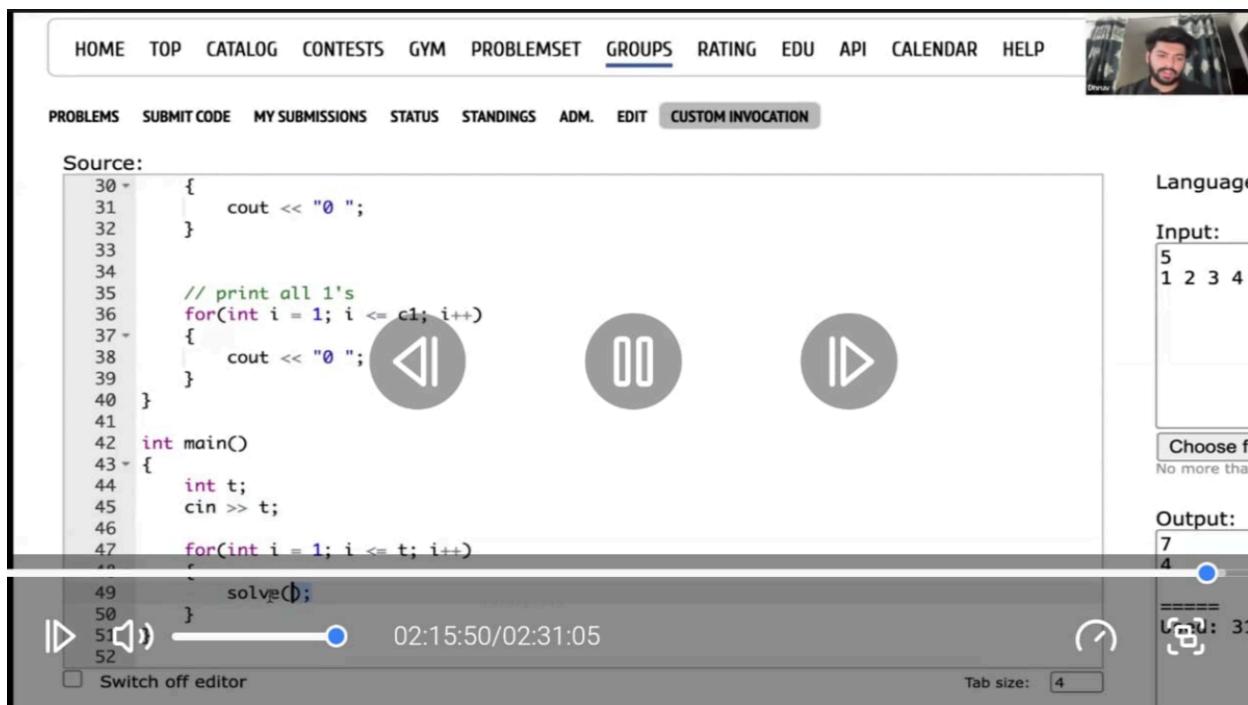
Language
Input:
5
1 2 3 4

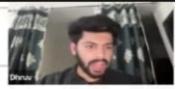
Choose fi
No more than

Output:
7
4

02:15:50/02:31:05

Switch off editor Tab size: 4



HOME TOP CATALOG CONTESTS GYM PROBLEMSET **GROUPS** RATING EDU API CALENDAR HELP 

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS STANDINGS ADM. EDIT **CUSTOM INVOCATION**

Source:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 void solve()
5 {
6     int n;
7     cin >> n;
8
9     int a[n];
10    for(int i = 0; i < n; i++)
11    {
12        cin >> a[i];
13    }
14
15    int c0 = 0, c1 = 0;
16    for(int i = 0; i < n; i++)
17    {
18        if(a[i] == 0)
19        {
20            c0++;
21        }
22    }
23 }
```

Language:

Input:
Choose file...
No more than 1 MB

Output:
Copy: 31

02:15:41/02:31:05

Switch off editor Tab size: 4