

1.1.You are given a sequence of  $n-1$  distinct positive integers, all of which are less than or equal to a integer 'n'. You have to find the integer that is missing from the range  $[1, 2, \dots, n]$ . Solve the question without using arrays.

Input Format:

One line containing the integer 'n' where  $2 \leq n \leq 10,000$

First line is followed by a sequence of ' $n-1$ ' distinct positive integers. Note that the sequence may not be in any particular order.

Output Format:

One line containing the missing number

Sample Test Cases

Test Case 1

Input

3

1 2

Output

3

Test Case 2

Input

4

1 3 4

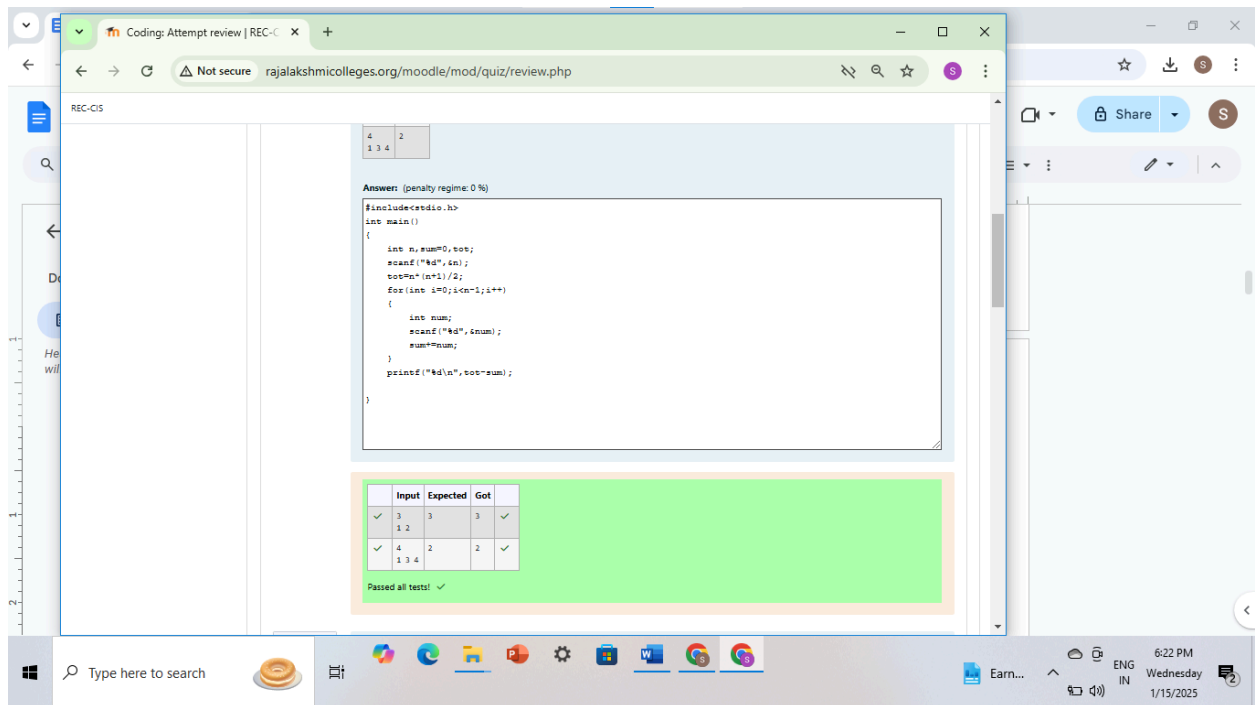
Output

2

**For example:**

Input	Result
3 1 2	3
4 1 3 4	2

**Answer:(penalty regime: 0 %)**



1.2.You are given a sequence of  $n-1$  distinct positive integers, all of which are less than or equal to a integer 'n'. You have to find the integer that is missing from the range  $[1, 2, \dots, n]$ . Solve the question without using arrays.

**Input Format:**

One line containing the integer 'n' where  $2 \leq n \leq 10,000$

First line is followed by a sequence of 'n-1' distinct positive integers. Note that the sequence may not be in any particular order.

**Output Format:**

One line containing the missing number

**Sample Test Cases**

**Test Case 1**

**Input**

3

1 2

**Output**

3

## Test Case 2

Input

4

1 3 4

Output

2

For example:

Input	Result
3 1 2	3
4 1 3 4	2

Answer:(penalty regime: 0 %)

The screenshot shows a web browser window displaying a coding attempt review page. The URL is [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). The page title is "Coding: Attempt review | REC-CIS".

The main content area displays the C code submitted by the user:

```
#include<stdio.h>
int main()
{
    int n,sum=0,tot;
    scanf("%d",&n);
    tot=n*(n+1)/2;
    for(int i=0;i<n-1;i++)
    {
        int num;
        scanf("%d",&num);
        sum+=num;
    }
    printf("%d\n",tot-sum);
}
```

Below the code, there is a table showing the test results:

Input	Expected	Got
3 1 2	3	3
4 1 3 4	2	2

The table indicates that all tests passed. Below the table, it says "Passed all tests! ✓".

At the bottom, there is a question description: "Question 2: A Teacher came to the class with a large box that has several coins. Each coin has a number Printed on it. Before Coming to the class, she ensured that all the coins occurs an even number of times. However, while coming to the class one coin fell down and got lost. She wants to find out the number of missing coin. Solve the question without using array."

1.2.A Teacher came to the class with a large box that has several coins. Each coin has a number Printed on it. Before Coming to the class, she ensured that all the coins occurs an even number of times. However, while coming to the class one coin fell down and got lost. She wants to find out the number of missing coin (Solve the question without using arrays).

Input Format:

Take Number from stdin which no of coins n.

Take n-1 array of Integers from stdin.

Output Format:

Print the number of coin which is missed.

Example Input:

8

5 7 2 7 5 2 5

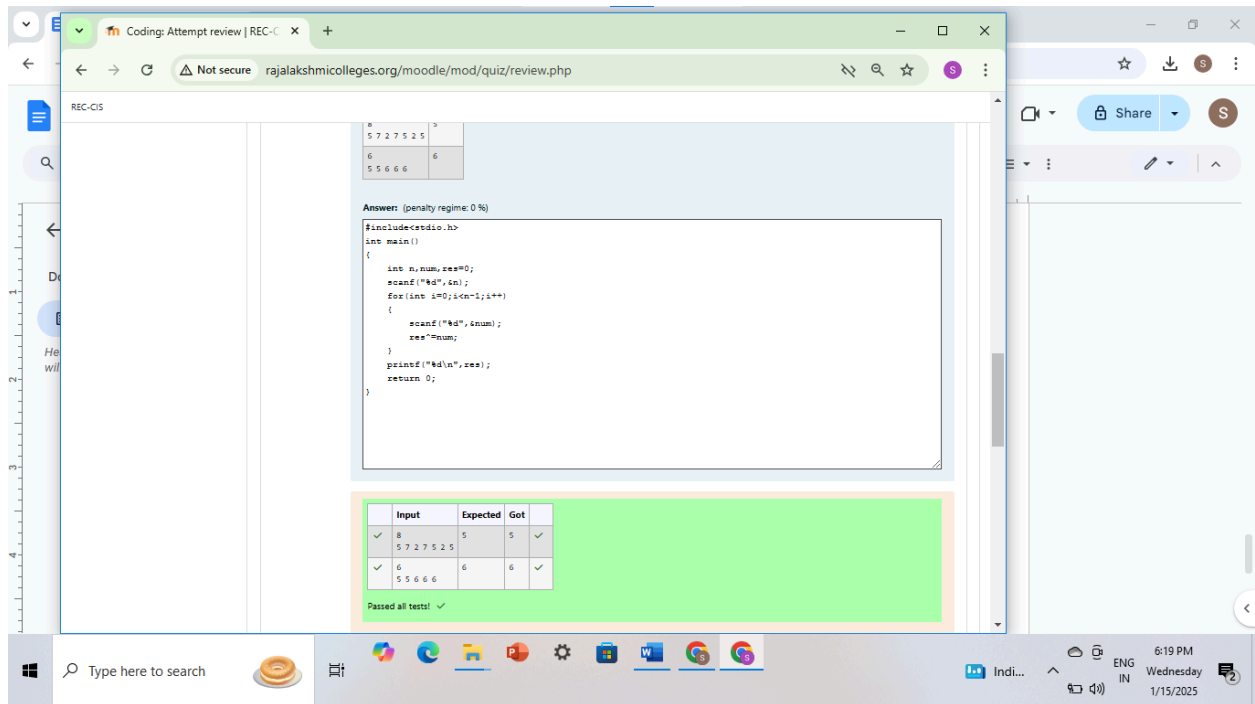
Output:

5

**For example:**

Input	Result
8 5 7 2 7 5 2 5	5
6 5 5 6 6 6	6

**Answer:**(penalty regime: 0 %)



1.3. An abundant number is a number for which the sum of its proper divisors is greater than the number itself.

Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Print Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is  $1 + 2 + 3 + 4 + 6 = 16$ . Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

**For example:**

Input	Result
12	Yes
13	No

**Answer:**(penalty regime: 0 %)

The screenshot shows a web browser window displaying a coding attempt review page. The page title is "Coding: Attempt review | REC-C". The URL is "rajalakshmicolleges.org/moodle/mod/quiz/review.php". The page content shows a C program that checks if a number is abundant. The program includes a function to calculate the sum of proper divisors and a main function that reads an integer and prints "Yes" or "No" based on whether the sum is greater than the number. Below the code, there is a table showing the test cases:

Input	Expected	Got
12	Yes	Yes
13	No	No

Below the table, it says "Passed all tests! ✓". At the bottom of the page, there is a "Finish review" button. The Windows taskbar is visible at the bottom of the screen, showing the time as 6:20 PM on Wednesday, 1/15/2025.

2.Sample Input 1:

5

30 40 50 20 10

20

Sample Output 1:

Element found at location : 3

Sample Input 2:

5

30 40 50 20 10

55

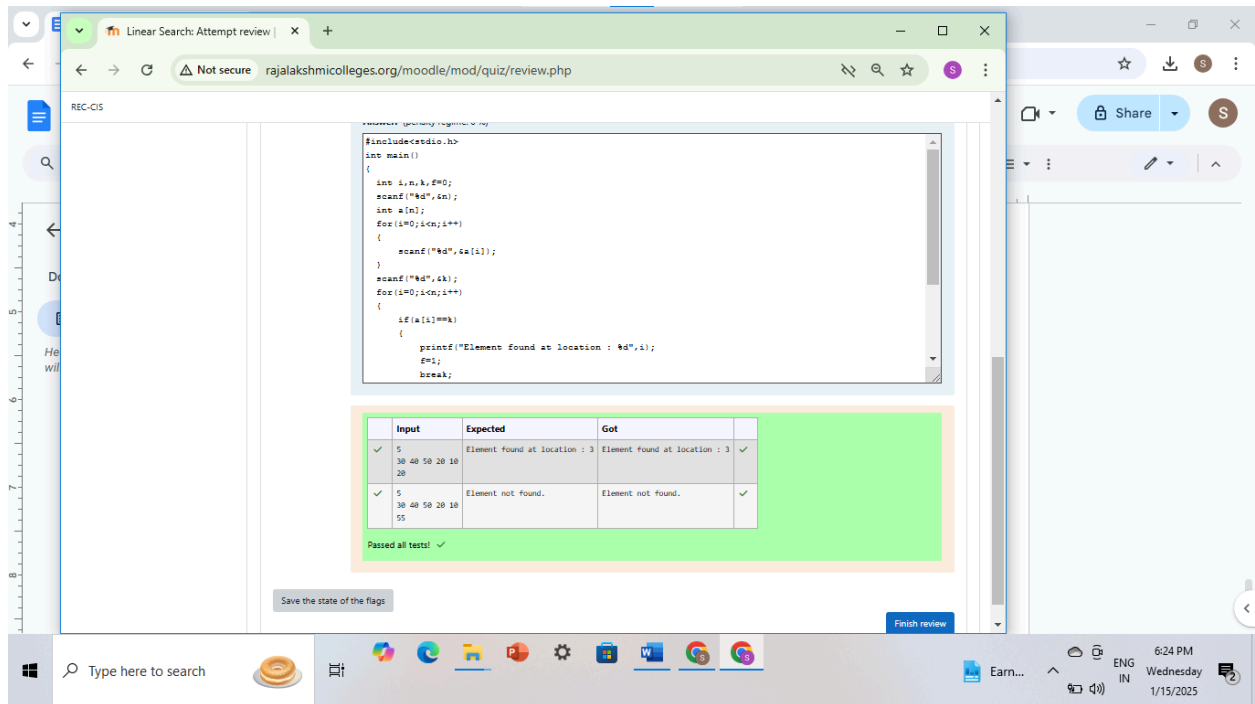
Sample Output 2:

Element not found.

**For example:**

Input	Result
5 30 40 50 20 10 20	Element found at location : 3
5 30 40 50 20 10 55	Element not found.

**Answer:**(penalty regime: 0 %)



3.Sample Input 1:

5

10 20 30 40 50

30

Sample Output 1:

Element found at location : 2

Sample Input 2:

5

10 20 30 40 50

55

Sample Output 2:

Element not found.

For example:

Input	Result
-------	--------



5 10 20 30 40 50 30	Element found at location : 2
5 10 20 30 40 50 55	Element not found.

## Answer

Binary Search: Attempt review | X +

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php

REC-CIS

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int binarySearch(int arr[],int n,int key){
    int low=0,high=n-1;
    while(low<=high){
        int mid = (low + high)/2;
        if(arr[mid]==key)
        {
            return mid;
        }
        else if(arr[mid]<key)
        {
            low=mid+1;
        }
        else
        {
            high=mid-1;
        }
    }
    return -1;
}
```

Input	Expected	Got	
✓ 5 10 20 30 40 50 30	Element found at location : 2	Element found at location : 2	✓
✓ 5 10 20 30 40 50 55	Element not found.	Element not found.	✓

Passed all tests! ✓

Windows taskbar: Type here to search, 28°C, ENG IN, 6:25 PM Wednesday 1/15/2025

### 4.Sample Input:

5  
30 40 50 20 10

### Sample Output:

10 20 30 40 50

### For example:

Input	Result
5 30 40 50 20 10	10 20 30 40 50

## Answer

REC-CIS

30 40 50 20 10

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    int arr[n];
    for(int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    for(int i = 1; i < n; i++)
    {
        int key = arr[i];
        int j = i - 1;
        while(j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j--;
        }
    }
}
```

	Input	Expected	Got	
X	5 30 40 50 20 10	10 20 30 40 50	30 40 50 20 10	X

Your code must pass all tests to earn any marks. Try again.

Show differences

Save the state of the flags

28°C ENG IN Wednesday 1/15/2025 6:27 PM

5.Sample Input:

5  
30 40 50 20 10

Sample Output:

10 20 30 40 50

For example:

Input	Result
5 30 40 50 20 10	10 20 30 40 50

--	--

**Answer:**(penalty regime: 0 %)

The screenshot shows a web browser window displaying a Moodle quiz review page. The URL is `rajalakshmicolleges.org/moodle/mod/quiz/review.php`. The page title is "Selection Sort: Attempt review". The quiz question is "REC-CIS". The answer is displayed as "Answer: (penalty regime: 0 %)". Below the answer, a C program for selection sort is shown in a code editor. The program includes `<stdio.h>`, defines `main()`, and implements the selection sort algorithm. Below the code, a table shows the test results:

Input	Expected	Got
5 30 40 50 20 10	10 20 30 40 50	10 20 30 40 50 ✓

Below the table, it says "Passed all tests! ✓". At the bottom of the quiz area, there are buttons for "Save the state of the flags" and "Finish review". The Windows taskbar is visible at the bottom of the screen, showing the time as 6:28 PM on Wednesday, 1/15/2025.

6.Sample Input:

5  
30 40 50 20 10

Sample Output:

10 20 30 40 50

**For example:**

Input	Result
5 30 40 50 20 10	10 20 30 40 50

**Answer:(penalty regime: 0 %)**

The screenshot shows a web browser window with the address bar displaying "Bubble Sort: Attempt review | R" and "Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php". The main content area displays a C program for bubble sort and its test results.

**Answer: (penalty regime: 0 %)**

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                int temp = arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
}
```

Input	Expected	Got	
✓ 5	10 20 30 40 50	10 20 30 40 50	✓
30 40 50 20 10			

Passed all tests! ✓

Save the state of the flags

Finish review

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with a temperature of 28°C, language set to ENG IN, and the date Wednesday 1/15/2025.