Q1i) Take a number, reverse it and add it to the original number until the obtained number is a palindrome.

**Constraints**

1<=num<=99999999

**Sample Input 1**

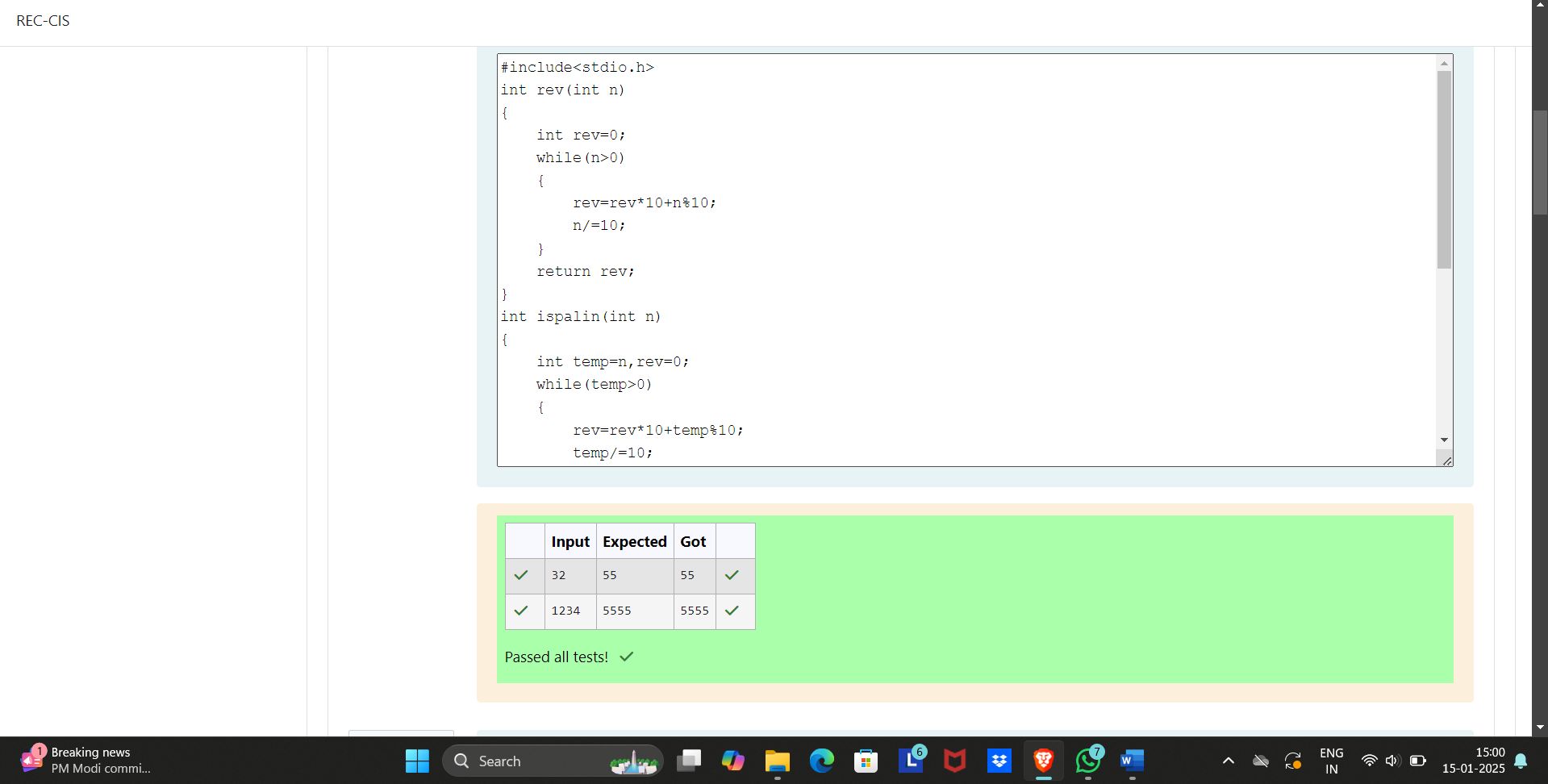
32

**Sample Output 1**

55

**For example:**

| **Input** | **Result** |
| --- | --- |
| 32 | 55 |
| 1234 | 5555 |



ii) Write a program to find if a given number N can be expressed as a sum of two prime numbers.

Note: YOU MUST OPTIMIZE the logic to find whether a number is prime or not, as very large prime numbers are provided as input. If the logic is not optimized your program will NOT get executed within the given time limit.

**Input Format:**

First line contains total number of test cases, denoted by T.

Next T lines will contain the value of N for each test case.

**Output Format:**

T lines containing either yes or no.

**Boundary Conditions / Constraints:**

1 <= T <= 25

3 <= N <= 10^9

**Example Input/Output 1:**

**Input:**

5

20

12

23

34

16

**Output:**

yes

yes

no

yes

yes

**Explanation:**

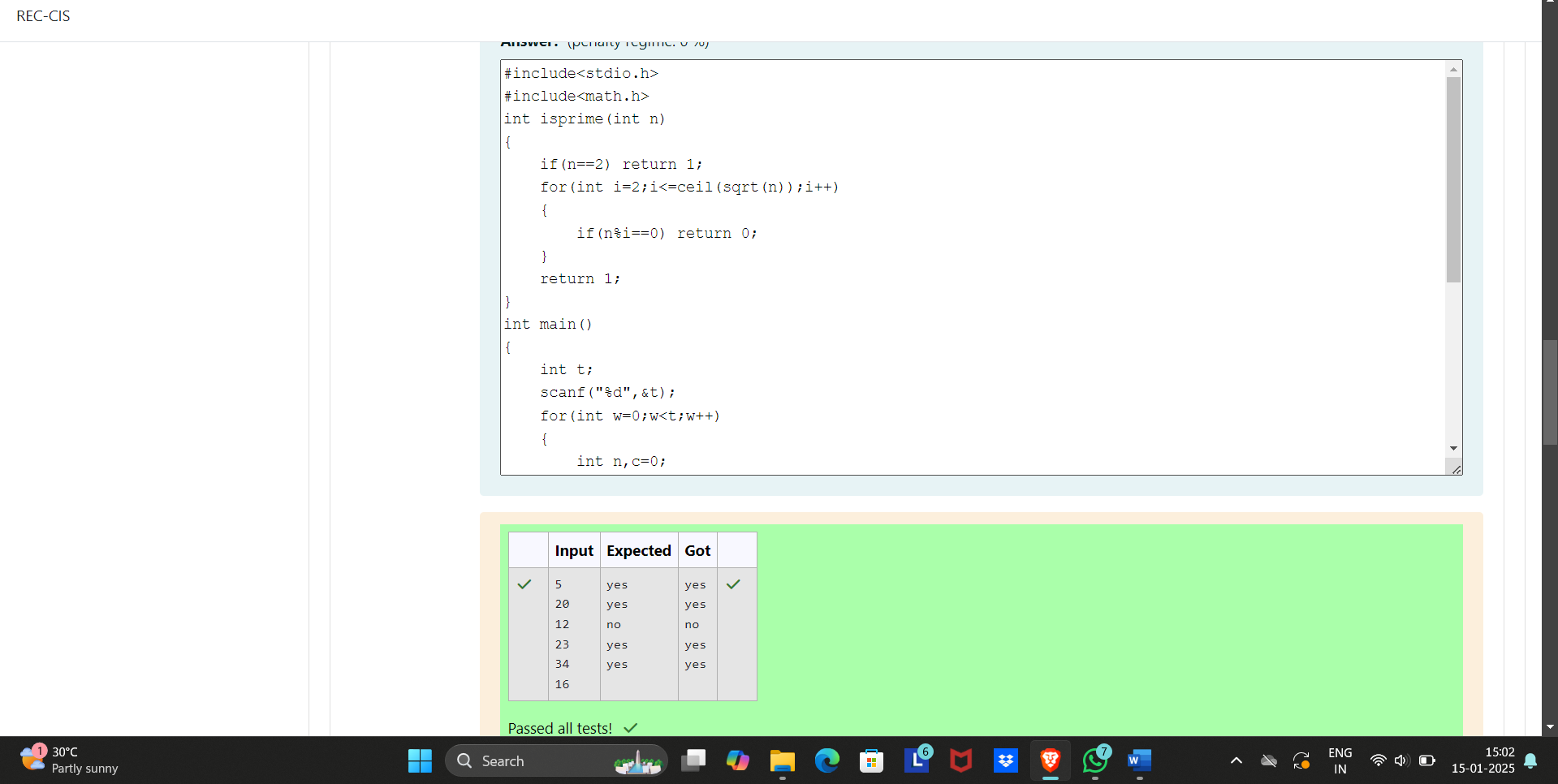
20 can be expressed as 17+3

12 can be expressed as 7+5

23 cannot be expressed as sum of two primes

34 can be expressed as 31+3 or 11+23 or 17+17

16 can be expressed as 11+5



iii) Write a C program that given an integer ‘n’, prints the number of integers that are less than or equal to ‘n’ and co-prime to ‘n’

Two integers a and b are said to be relatively prime or co-prime if the only positive integer that evenly divides both of them is 1. That is, the only common positive factor of the two numbers is 1. This is equivalent to their greatest common divisor being 1.

Input Format:

One line containing the value of 'n', where 1<=n<=10,000

Output Format:

One line containing the number of integers that are co-prime to n and less than or equal to 'n'

Sample Test Cases

Test Case 1

Input

10

Output

4

Test Case 2

Input

23

Output

22

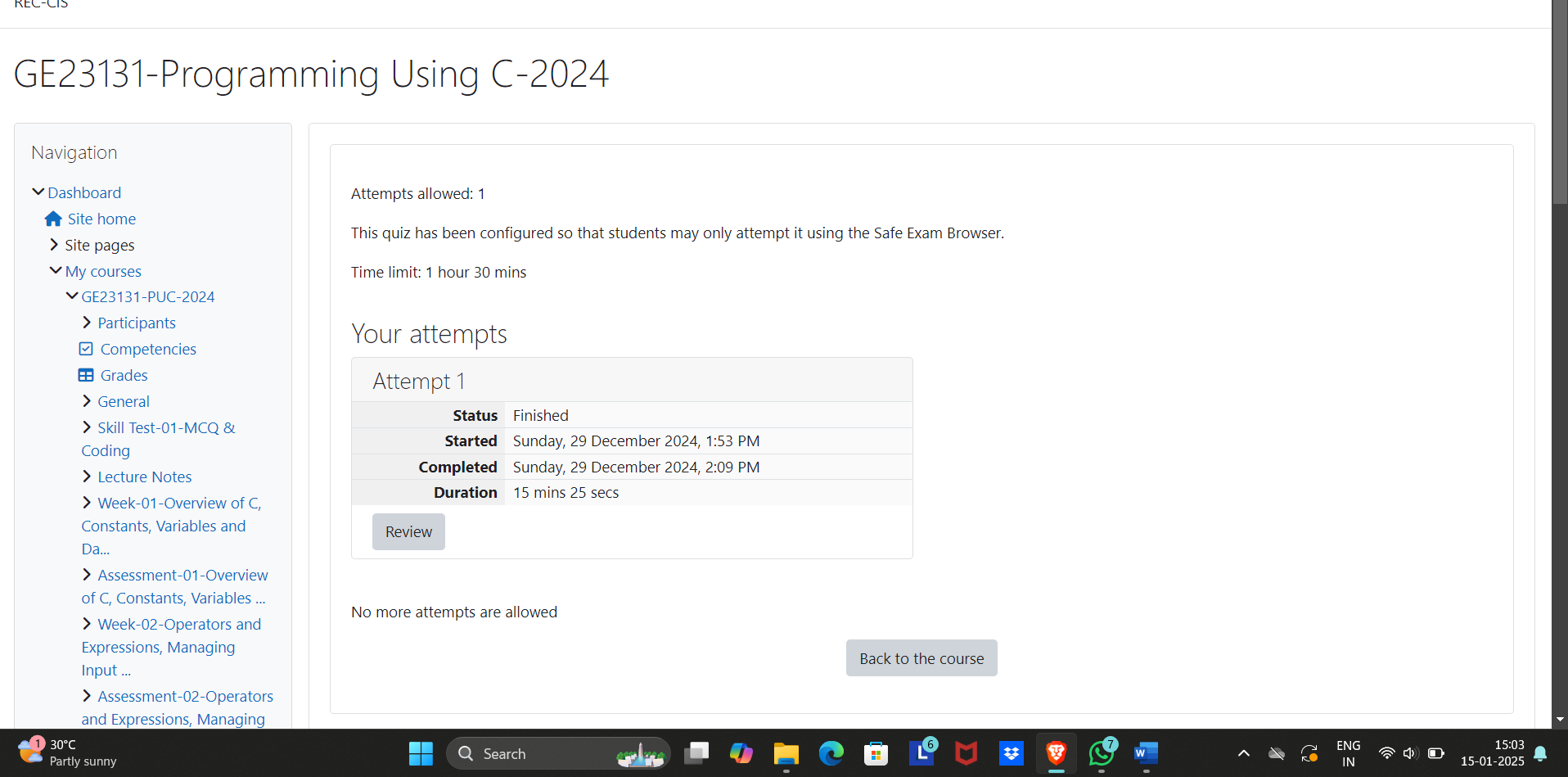
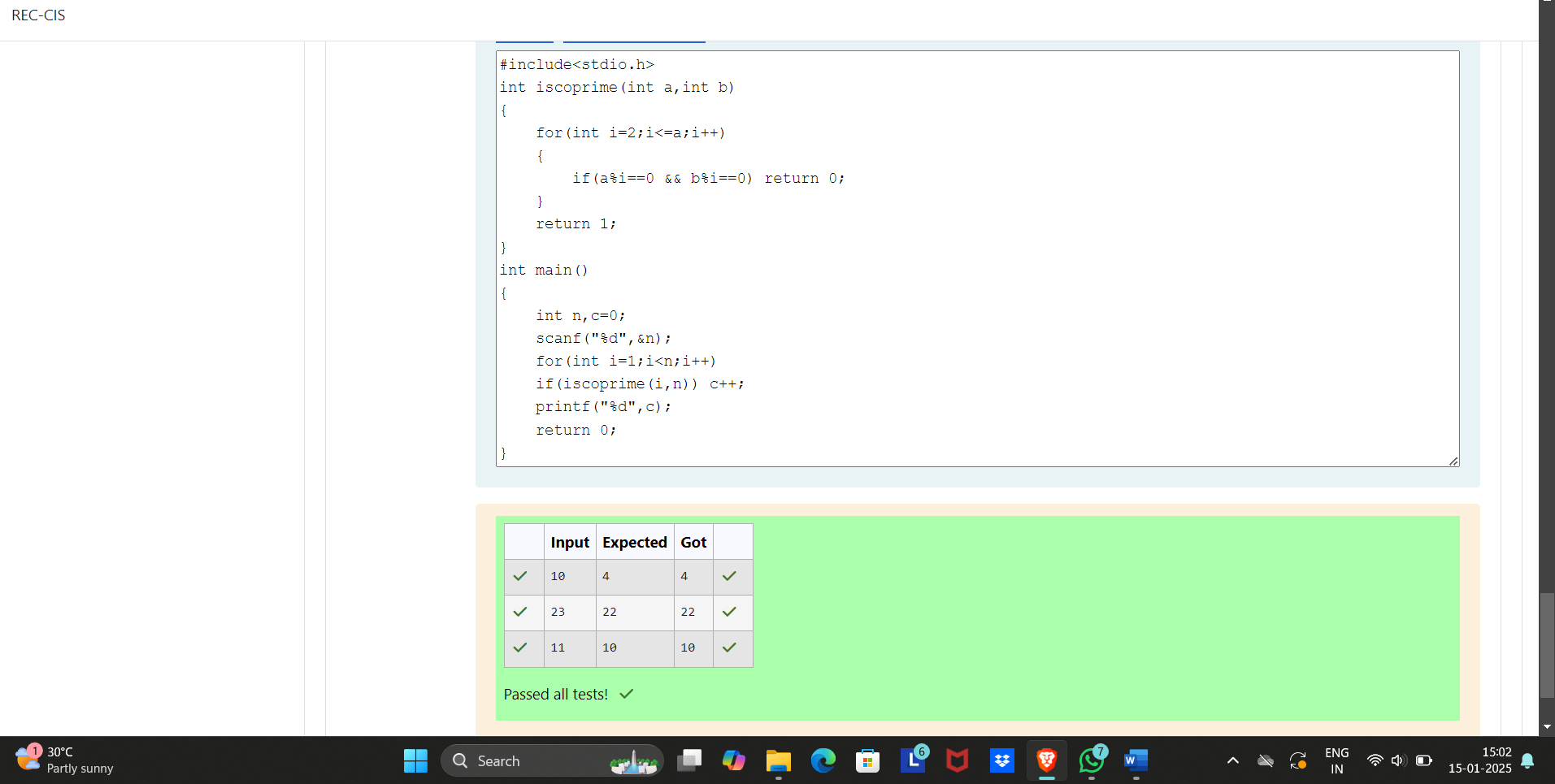
Test Case 3

Input

11

Output

10



Q2i) Coders here is a simple task for you, you have given an array of size ***N*** and an integer ***M***.

Your task is to calculate the ***difference between maximum sum and minimum sum of N-M*** elements of the given array.

**Constraints:**

***1<=t<=10***

***1<=n<=1000***

***1<=a[i]<=1000***

**Input:**

First line contains an integer ***T*** denoting the number of testcases.

First line of every testcase contains two integer ***N*** and ***M***.

Next line contains ***N*** space separated integers denoting the elements of array

**Output:**

For every test case print your answer in new line

SAMPLE INPUT

1

5 1

1 2 3 4 5

SAMPLE OUTPUT

4

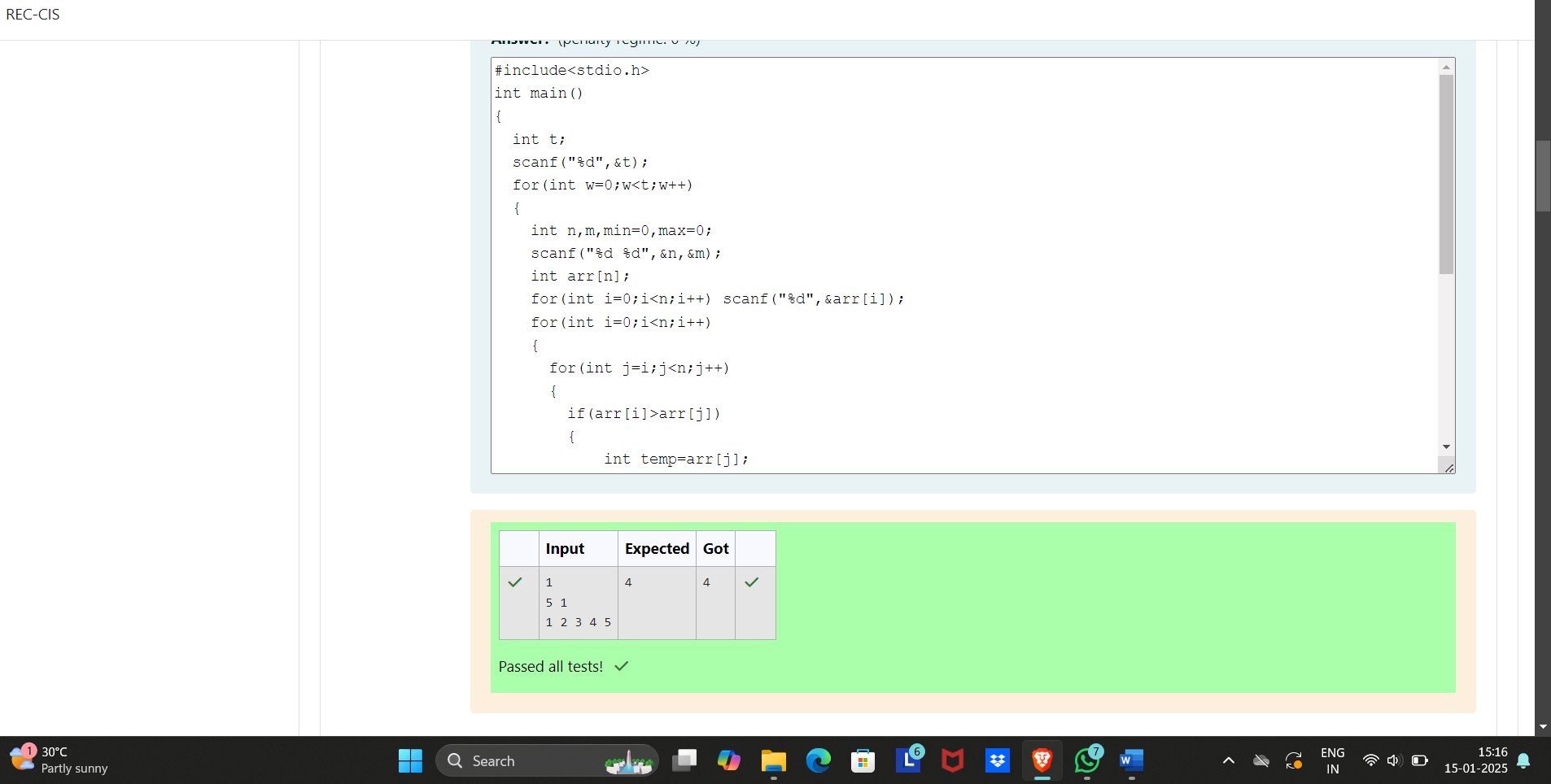
Explanation

M is 1 and N is 5 so you have to calculate maximum and minimum sum using (5-1 =) 4 elements.

Maximum sum using the 4 elements would be (2+3+4+5=)14.

Minimum sum using the 4 elements would be (1+2+3+4=)10.

Difference will be 14-10=4.



ii) A new deadly virus has infected large population of a planet. A brilliant scientist has discovered a new strain of virus which can cure this disease. Vaccine produced from this virus has various strength depending on midichlorians count. A person is cured only if midichlorians count in vaccine batch is more than midichlorians count of person. A doctor receives a new set of report which contains midichlorians count of each infected patient, Practo stores all vaccine doctor has and their midichlorians count. You need to determine if doctor can save all patients with the vaccines he has. The number of vaccines and patients are equal.

**Input Format**

First line contains the number of vaccines - N. Second line contains N integers, which are strength of vaccines. Third line contains N integers, which are midichlorians count of patients.

**Output Format**

Print a single line containing ***'Yes'*** or ***'No'***.

**Input Constraint**

***1 < N < 10***

Strength of vaccines and midichlorians count of patients fit in integer.

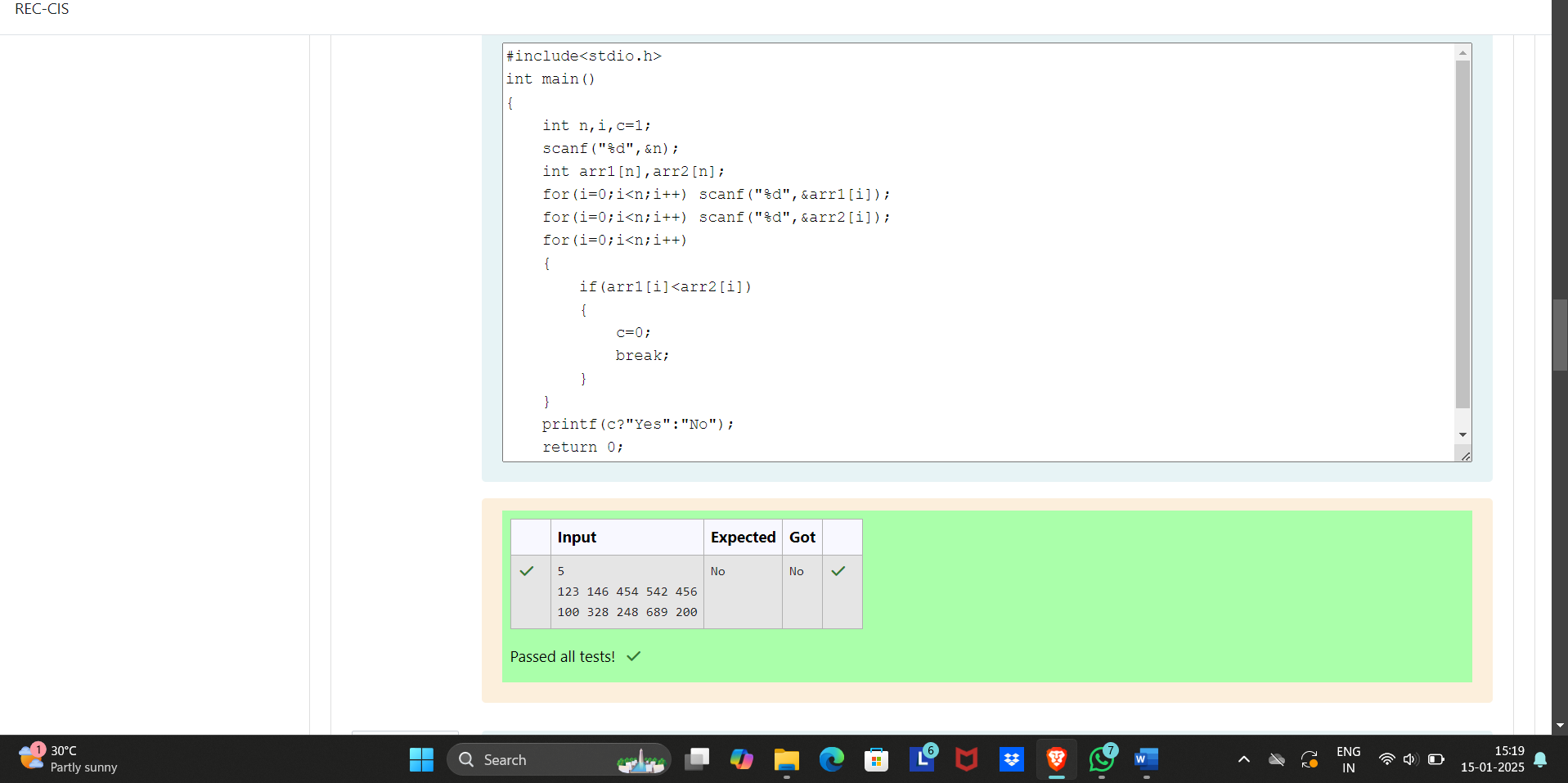
**SAMPLE INPUT**

5

123 146 454 542 456

100 328 248 689 200

**SAMPLE OUTPUT**

NoYou are given an array of n integer numbers ***a1, a2, . . . , an***. Calculate the number of pair of indices ***(i, j)*** such that ***1 ≤ i < j ≤ n*** and ***ai*** xor ***aj = 0***.

**Input format**

- First line: ***n*** denoting the number of array elements

- Second line: n space separated integers ***a1, a2, . . . , an***.

**Output format**

Output the required number of pairs.

**Constraints**

***1 ≤ n ≤ 106***

***1 ≤ ai ≤ 109***

**SAMPLE INPUT**

5

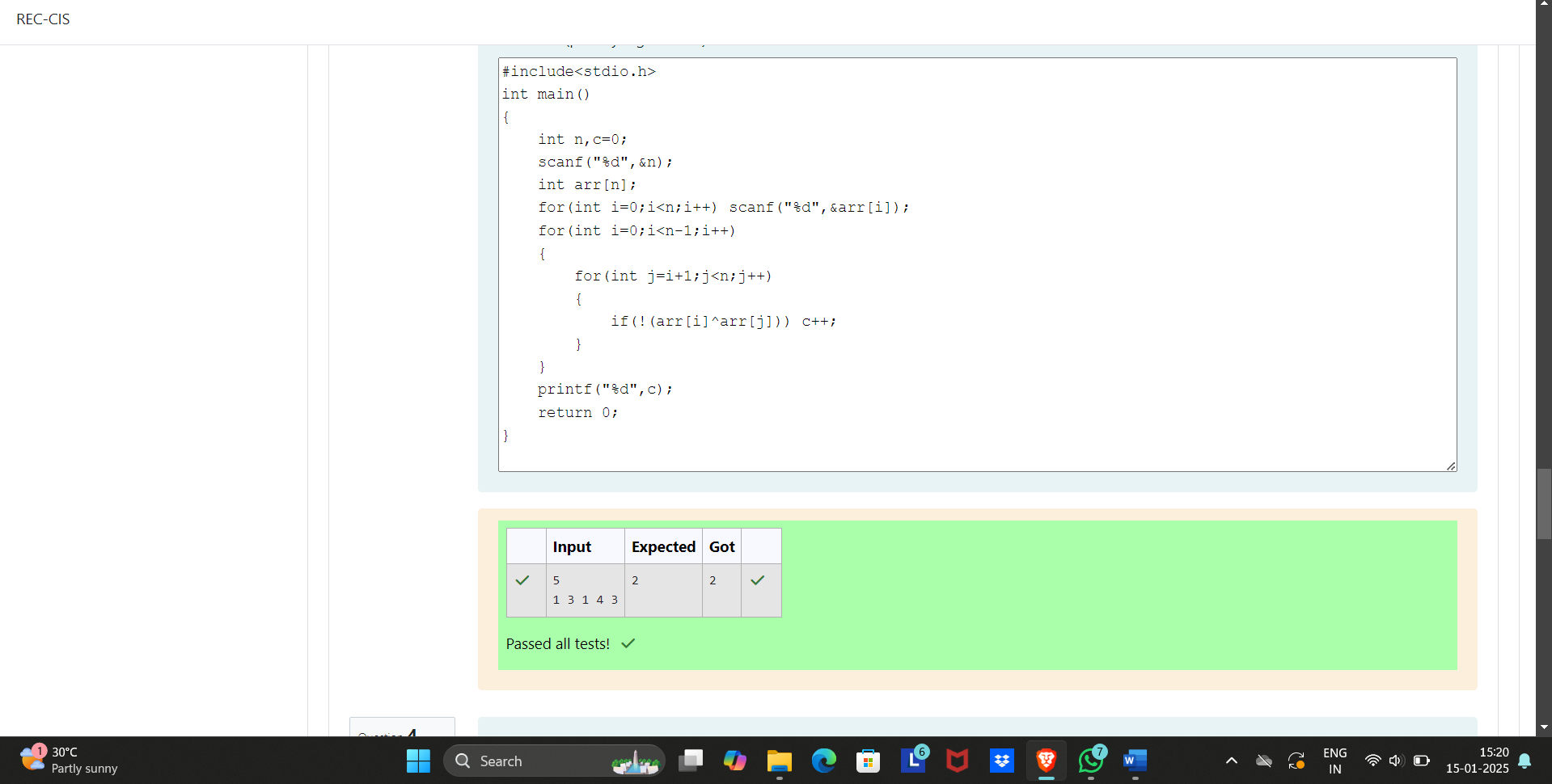
1 3 1 4 3

**SAMPLE OUTPUT**

2

Explanation

The 2 pair of indices are ***(1, 3)*** and ***(2,5)***.



iv) You are given an array ***A*** of non-negative integers of size ***m***. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

**Example:**

A={4,5,3,7,1}

After sorting the new array becomes A={1,3,4,5,7}.

The required output should be "4 2 0 1 3"

**INPUT :**

The first line of input consists of the size of the array

The next line consists of the array of size m

**OUTPUT :**

Output consists of a single line of integers

**CONSTRAINTS:**

***1<=m<=106***

***0<=A[i]<=106***

NOTE: The indexing of the array starts with 0.

**SAMPLE INPUT**

5

4 5 3 7 1

**SAMPLE OUTPUT**

4 2 0 1 3

