**Week 6**

**Question 1**

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that

A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:

2. N, followed by N integers of the array

3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn’t.

Example

Input:

1

3 1 3 5

4

Output:

1

Input:

1

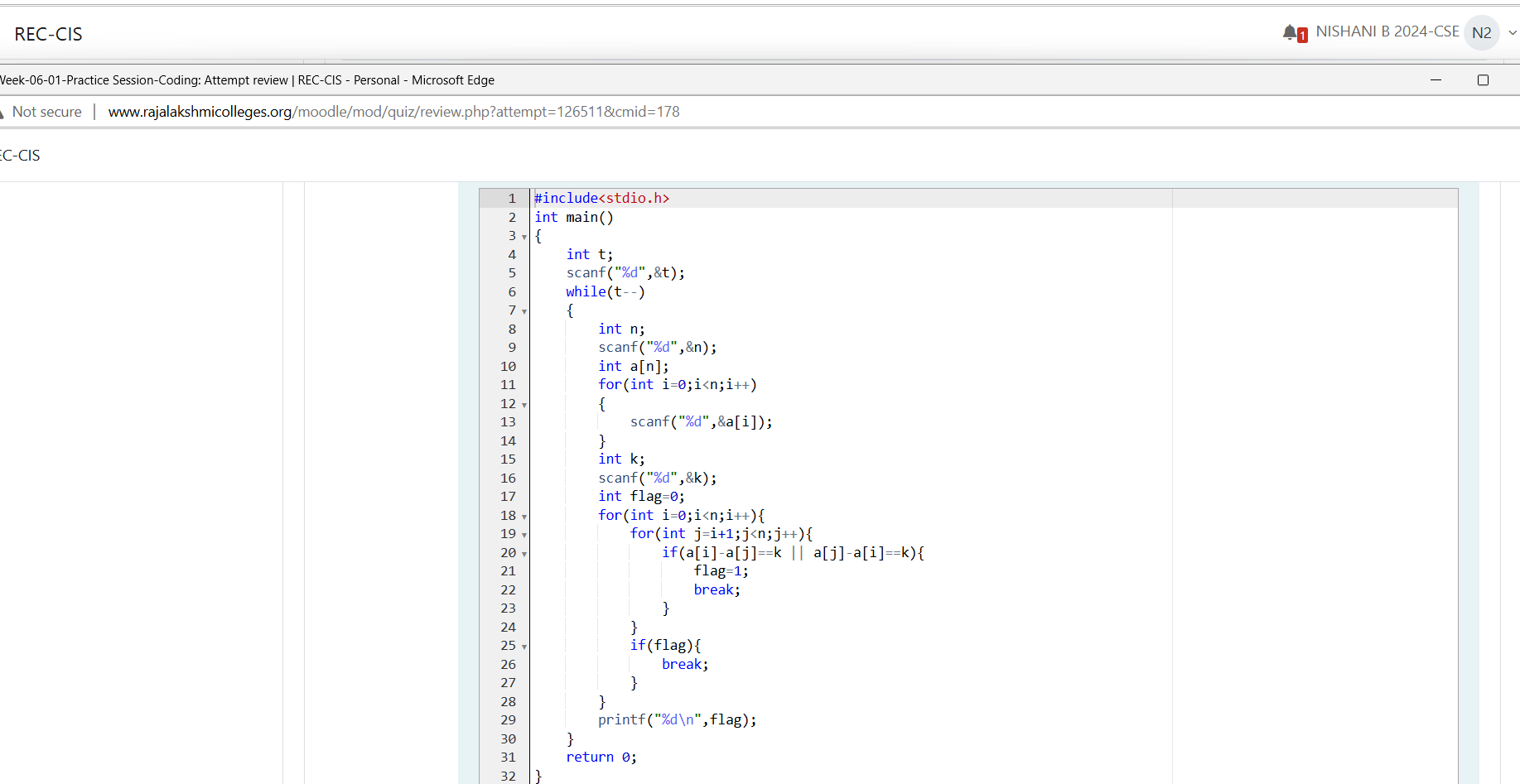
3 1 3 5

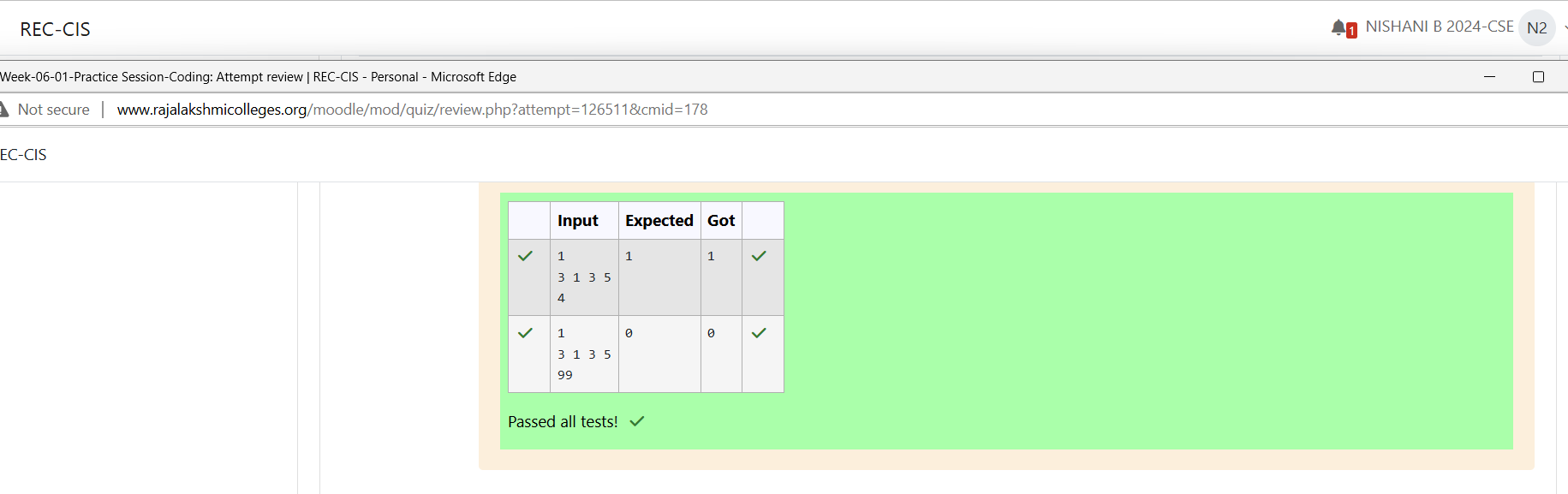
99

Output:

0

**PROGRAM:**



**OUTPUT:**

**Question 2**

Sam loves chocolates and starts buying them on the 1st day of the year. Each day of the year, x, is numbered from 1 to

Y. On days when x is odd, Sam will buy x chocolates; on days when x is even, Sam will not purchase any chocolates.

Complete the code in the editor so that for each day Ni (where 1 ≤ x ≤ N ≤ Y) in array arr, the number of chocolates

Sam purchased (during days 1 through N) is printed on a new line. This is a function-only challenge, so input is handled

for you by the locked stub code in the editor.

Input Format

The program takes an array of integers as a parameter.

The locked code in the editor handles reading the following input from stdin, assembling it into an array of integers

(arr), and calling calculate(arr).

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes

the ith test case as an integer, Ni (the number of days).

Constraints

1 ≤ T ≤ 2 × 105

1 ≤ N ≤ 2 × 106

1 ≤ x ≤ N ≤ Y

Output Format

For each test case, Ti in arr, your calculate method should print the total number of chocolates Sam purchased by day

Ni on a new line.

Sample Input 0

3

1

2

3

Sample Output 0

1

1

4

Explanation

Test Case 0: N = 1

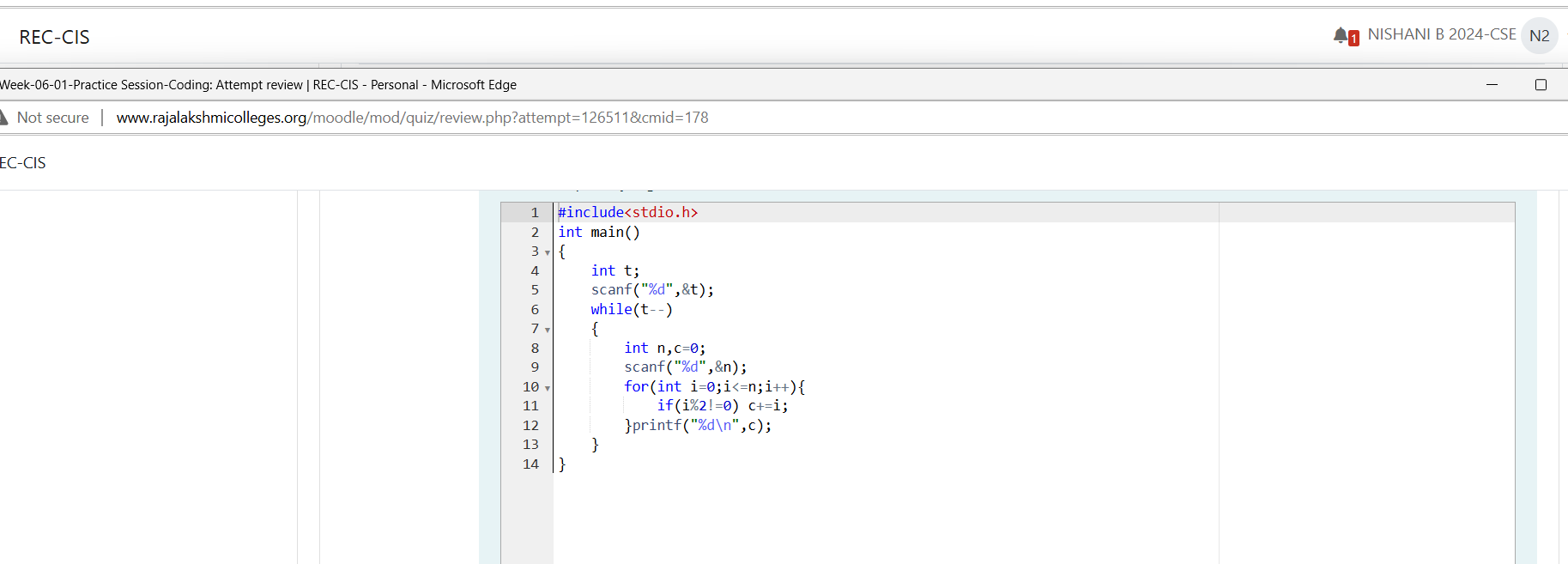
Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 1: N = 2

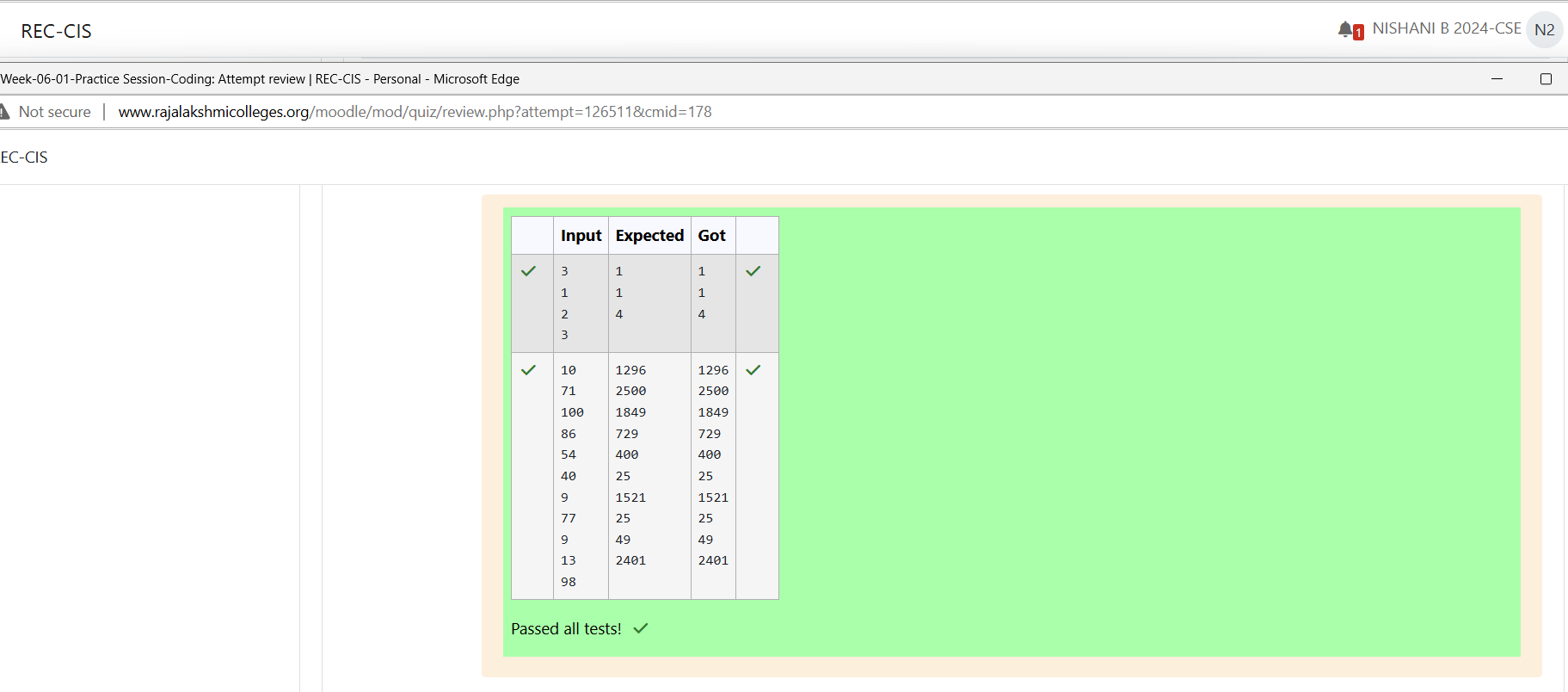
Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 2: N = 3

Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 on a new line

**PROGRAM**:

**OUTPUT:**

**Question 3**

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

• Football team A, has played three matches, and has scored { 1 , 2 , 3 } goals in each match respectively.

• Football team B, has played two matches, and has scored { 2, 4 } goals in each match respectively.

• Your task is to compute, for each match of team B, the total number of matches of team A, where team A has

scored less than or equal to the number of goals scored by team B in that match.

• In the above case:

• For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2.

• For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of m positive integers, one for each maxes[i]

representing the total number of elements nums[j] satisfying nums[j] ≤ maxes[i] where 0 ≤ j < n and 0 ≤ i < m, in the

given order.

It has the following:

nums[nums[0],...nums[n-1]]: first array of positive integers

maxes[maxes[0],...maxes[n-1]]: second array of positive integers

Constraints

• 2 ≤ n, m ≤ 105

• 1 ≤ nums[j] ≤ 109, where 0 ≤ j < n.

• 1 ≤ maxes[i] ≤ 109, where 0 ≤ i < m.

Input Format For Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number of elements in nums.

The next n lines each contain an integer describing nums[j] where 0 ≤ j < n.

The next line contains an integer m, the number of elements in maxes.

The next m lines each contain an integer describing maxes[i] where 0 ≤ i < m.

Sample Case 0

Sample Input 0

4

1

4

2

4

2

3

5

Sample Output 0

2 60

4

Explanation 0

We are given n = 4, nums = [1, 4, 2, 4], m = 2, and maxes = [3, 5].

1. For maxes[0] = 3, we have 2 elements in nums (nums[0] = 1 and nums[2] = 2) that are ≤ maxes[0].

2. For maxes[1] = 5, we have 4 elements in nums (nums[0] = 1, nums[1] = 4, nums[2] = 2, and nums[3] = 4) that are≤ maxes[1].

Thus, the function returns the array [2, 4] as the answer.

Sample Case 1

Sample Input 1

5

2

10

5

4

8

4

3

1

7

8

Sample Output 1

1

0

3

4

Explanation 1

We are given, n = 5, nums = [2, 10, 5, 4, 8], m = 4, and maxes = [3, 1, 7, 8].

1. For maxes[0] = 3, we have 1 element in nums (nums[0] = 2) that is ≤ maxes[0].

2. For maxes[1] = 1, there are 0 elements in nums that are ≤ maxes[1].

3. For maxes[2] = 7, we have 3 elements in nums (nums[0] = 2, nums[2] = 5, and nums[3] = 4) that are ≤ maxes[2].

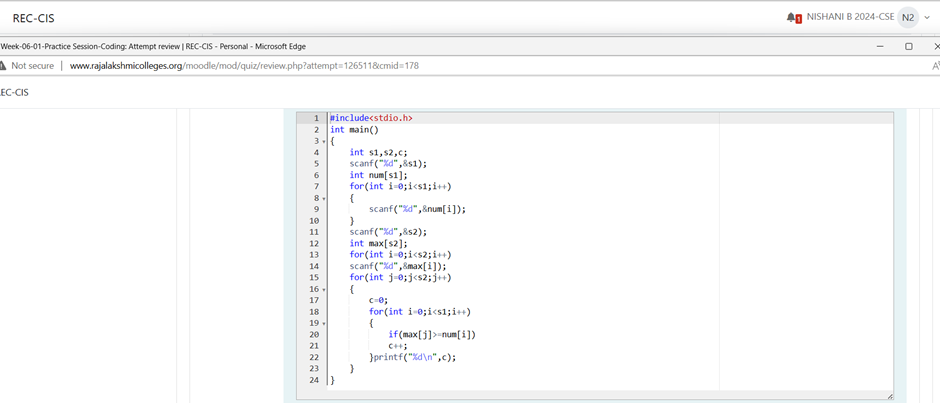
4. For maxes[3] = 8, we have 4 elements in nums (nums[0] = 2, nums[2] = 5, nums[3] = 4, and nums[4] = 8) that are

≤ maxes[3].

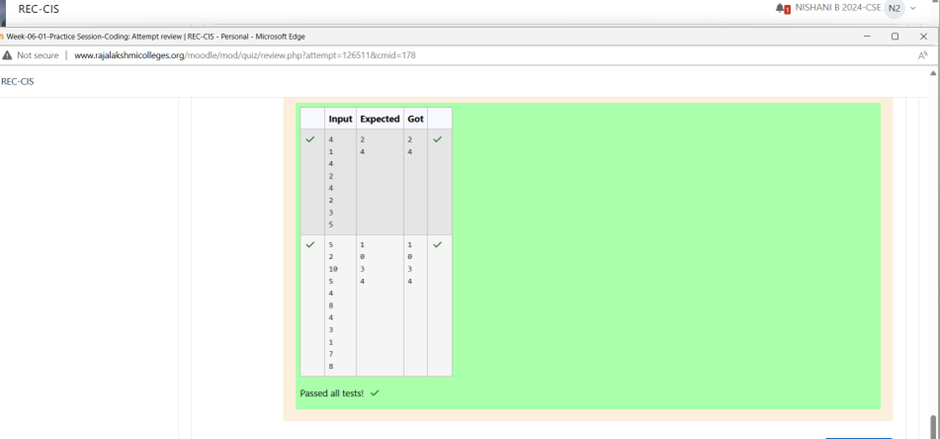
61

Thus, the function returns the array [1, 0, 3, 4] as the answer.

**PROGRAM:**



**OUTPUT:**



**WEEK 6**

**Question 1**

Given an array of numbers and a window of size k. Print the maximum of numbers inside the window for each step as

the window moves from the beginning of the array.

Input Format

Input contains the array size, no of elements and the window size

Output Format

Print the maximum of numbers

Constraints

1 <= size <= 1000

Sample Input 1

8

1 3 5 2 1 8 6 9

3

Sample Output 1

5 5 5 8 8 9

For example:

Input

8

1 3 5 2 1 8 6 9

3

result

5 5 5 8 8 9

input

10

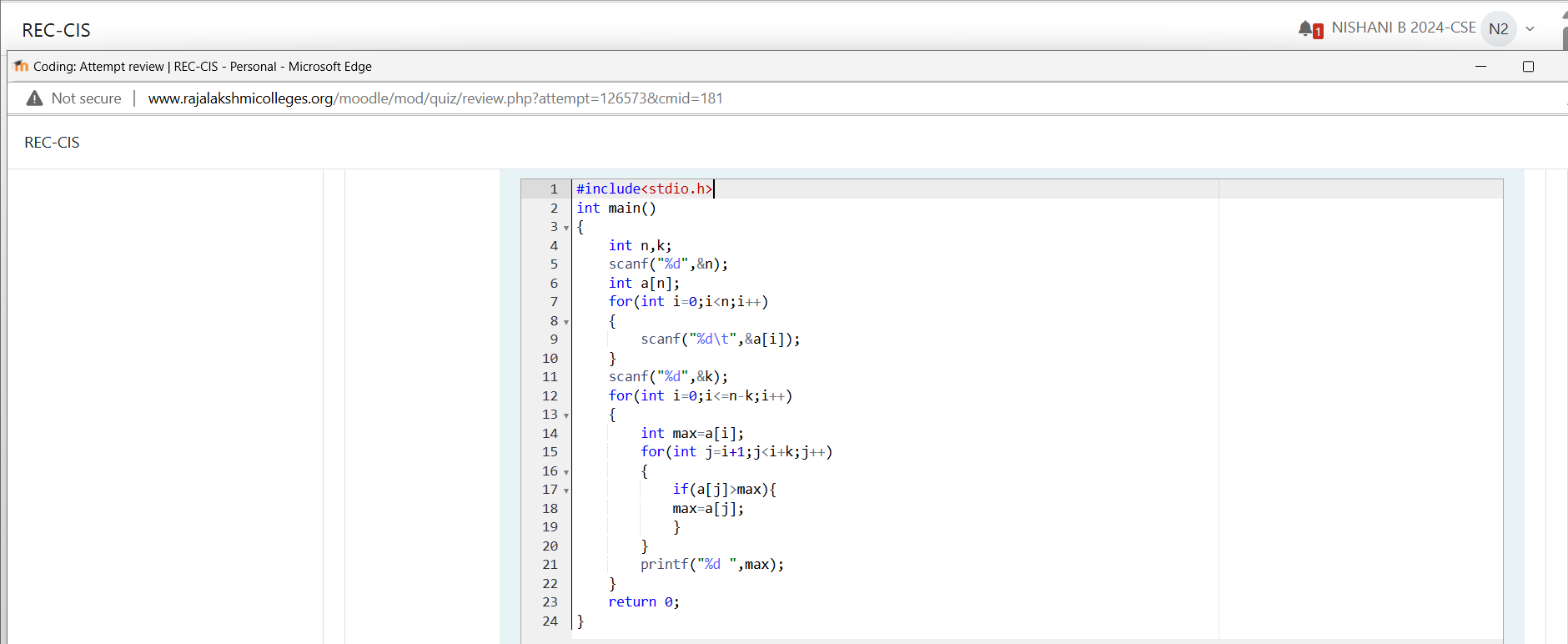
3 7 5 1 2 9 8 5 3 2

3

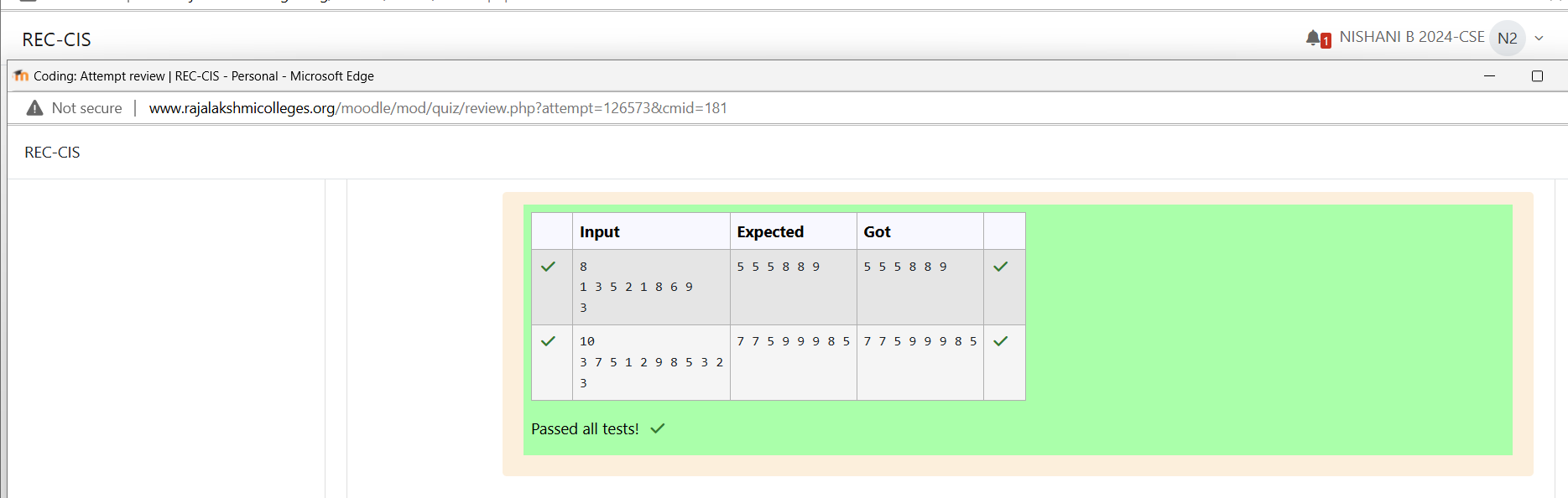
Result:

7 7 5 9 9 9 8 5

**PROGRAM:**



**OUTPUT:**



**Question 3**

Given an array and a threshold value find the output.

Input: {5,8,10,13,6,2}

Threshold = 3

Output count = 17

Explanation:

Number Parts Counts

5 {3,2} 2

8 {3,3,2} 3

10 {3,3,3,1} 4

13 {3,3,3,3,1} 5

6 {3,3} 2

2 {2} 1

Input Format

N - no of elements in an array

Array of elements

Threshold value

Output Format

Display the count

Sample Input 1

6

5 8 10 13 6 2

Sample Output1

17

For example:

Input

6

5 8 1 0 1 3 6 2 3

Result:

17

Input:

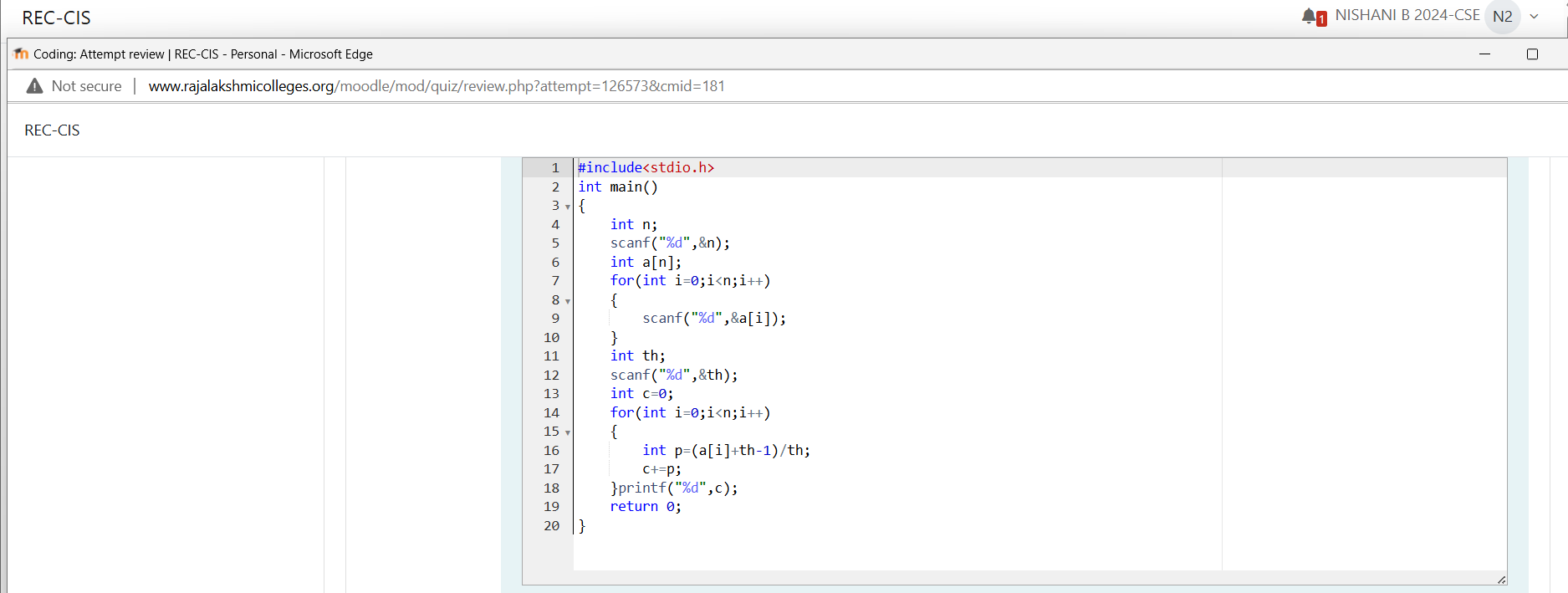
7

20 35 57 30 56 87 30 33

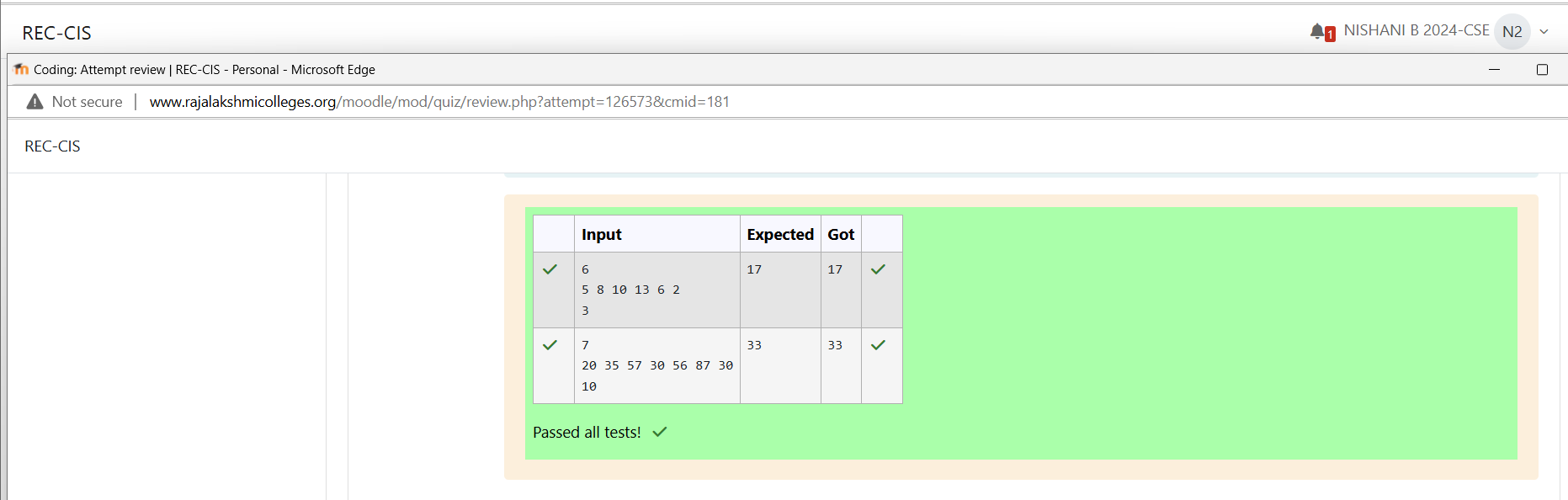
Result:

10

**PROGRAM**:



**OUTPUT:**



**Question 3:**

Output is an merged array without duplicates

Input format

N1-no of elements in array 1

Array elements for array 1

N2-no of elements in array 2

Array elements for array 2

Display the merged array

Sample input 1

5

1 2 3 6 9

4

2 4 5 10

Sample output 1

1 2 3 4 5 6 7 8 9 10

For example:

Input

5

1 2 3 6 9

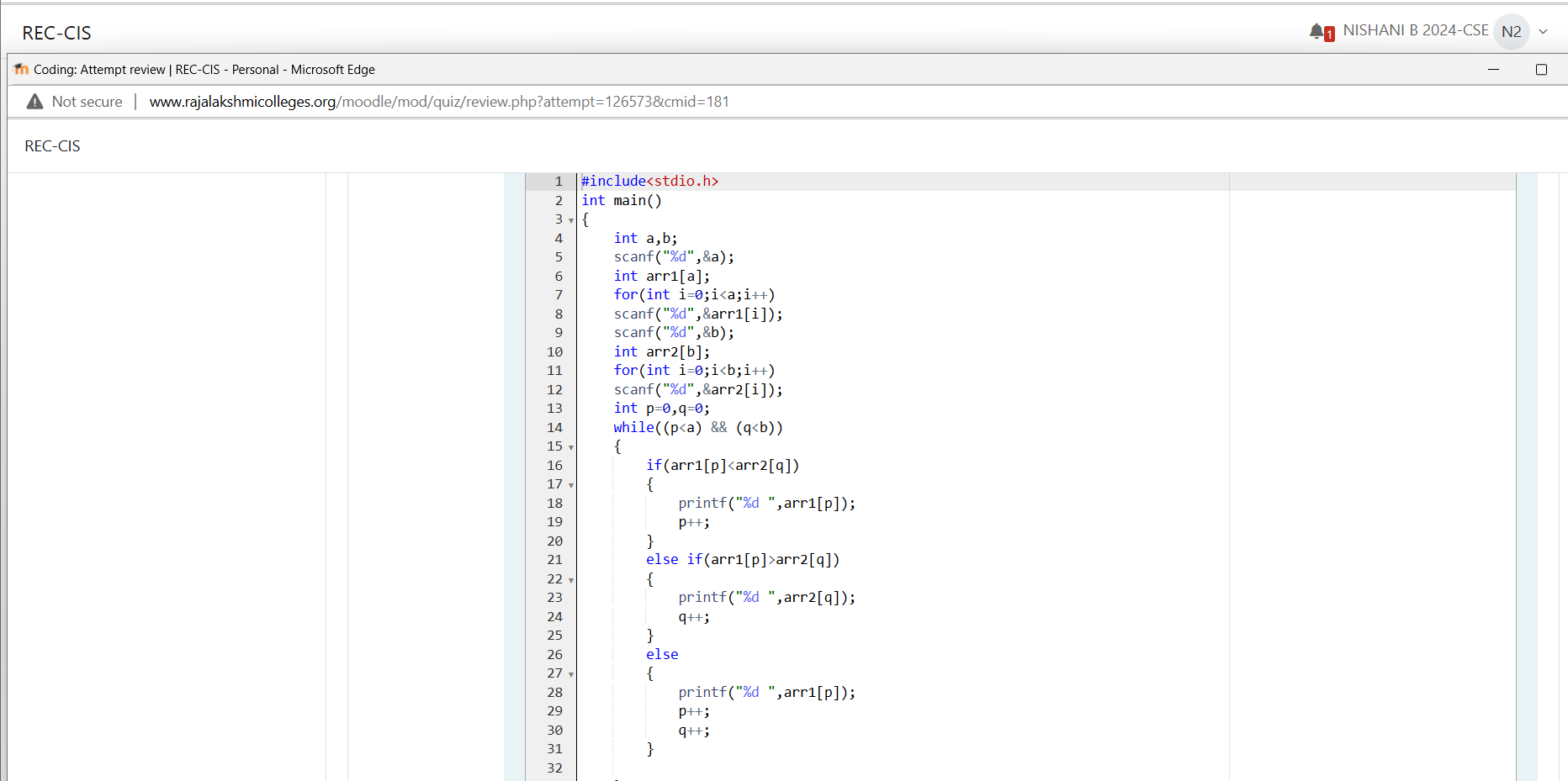
4

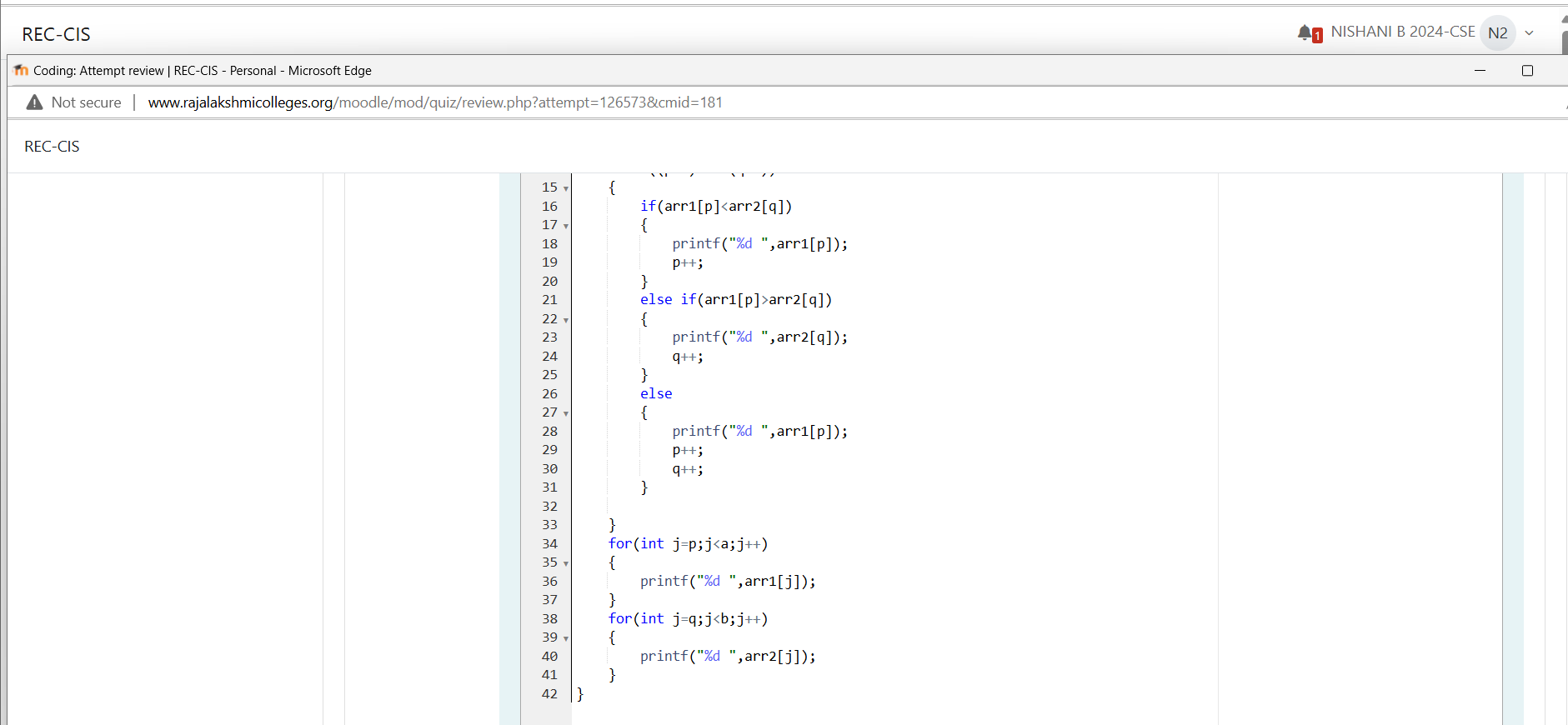
2 4 5 10

Result

1 2 3 4 5 6 9 10

**PROGRAM:**





**OUTPUT:**