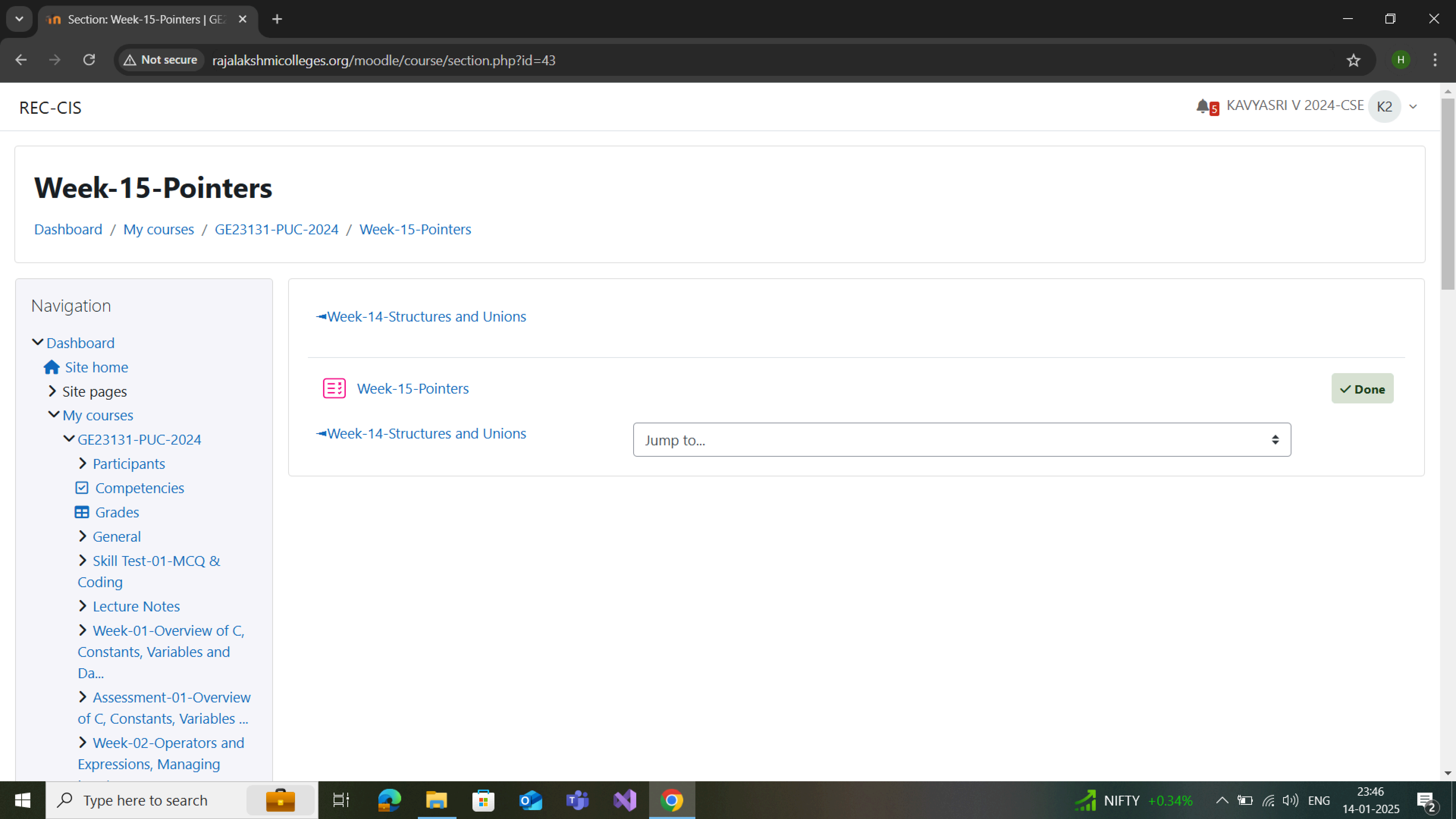

Section: Week-15-Pointers | GE

Not secure rajalakshmicolleges.org/moodle/course/section.php?id=43

REC-CIS

KAVYASRI V 2024-CSE K2

Week-15-Pointers

Dashboard / My courses / GE23131-PUC-2024 / Week-15-Pointers

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - GE23131-PUC-2024
      - Participants
      - Competencies
      - Grades
      - General
      - Skill Test-01-MCQ & Coding
      - Lecture Notes
      - Week-01-Overview of C, Constants, Variables and Da...
      - Assessment-01-Overview of C, Constants, Variables ...
      - Week-02-Operators and Expressions, Managing

◄Week-14-Structures and Unions

Week-15-Pointers ✓ Done

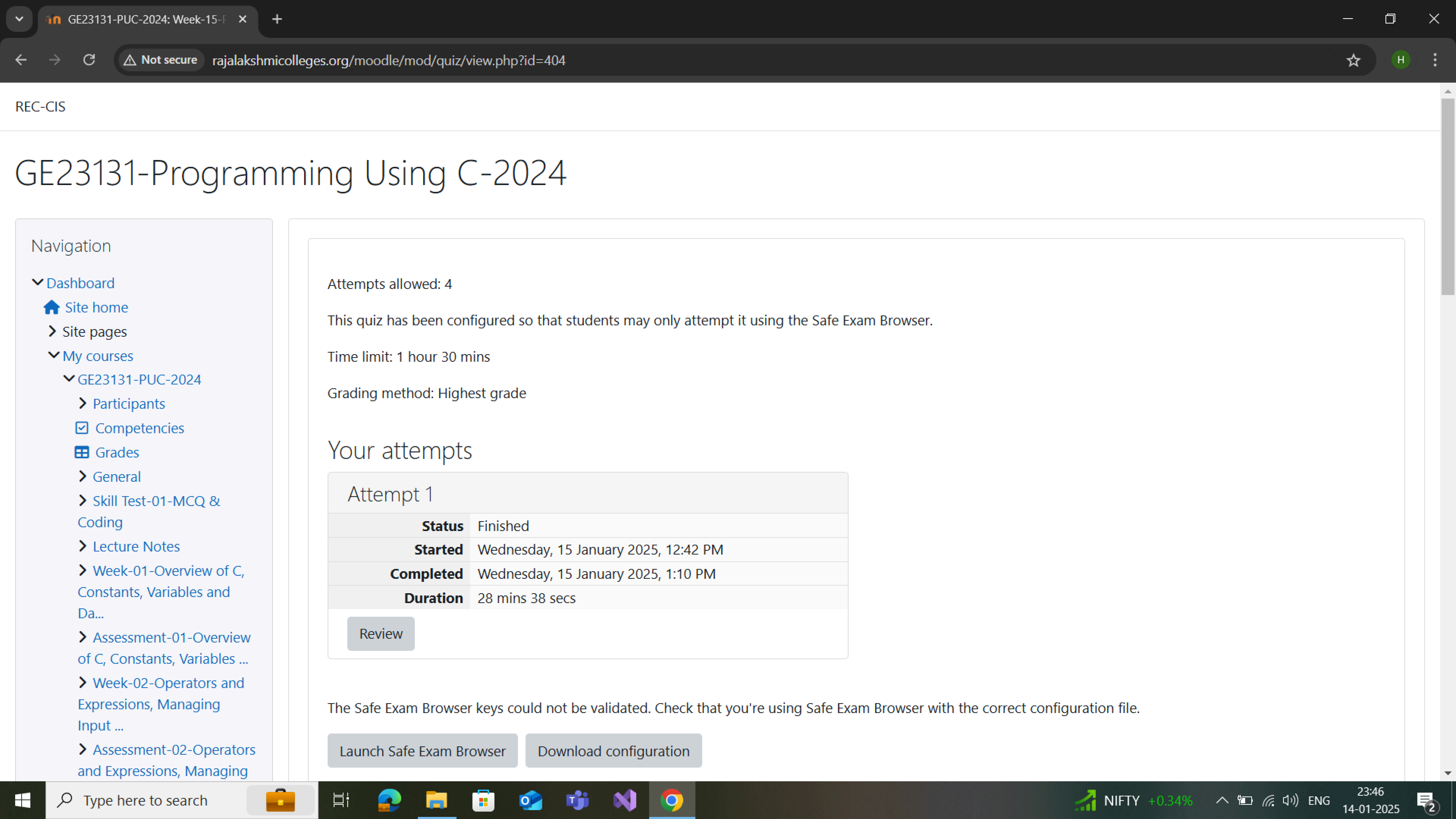
◄Week-14-Structures and Unions

Jump to...

Type here to search

NIFTY +0.34%

ENG 23:46 14-01-2025


GE23131-PUC-2024: Week-15-F

rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=404

REC-CIS

GE23131-Programming Using C-2024

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - GE23131-PUC-2024
      - Participants
      - Competencies
      - Grades
      - General
      - Skill Test-01-MCQ & Coding
      - Lecture Notes
      - Week-01-Overview of C, Constants, Variables and Da...
      - Assessment-01-Overview of C, Constants, Variables ...
      - Week-02-Operators and Expressions, Managing Input ...
      - Assessment-02-Operators and Expressions, Managing

Attempts allowed: 4

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 1 hour 30 mins

Grading method: Highest grade

Your attempts

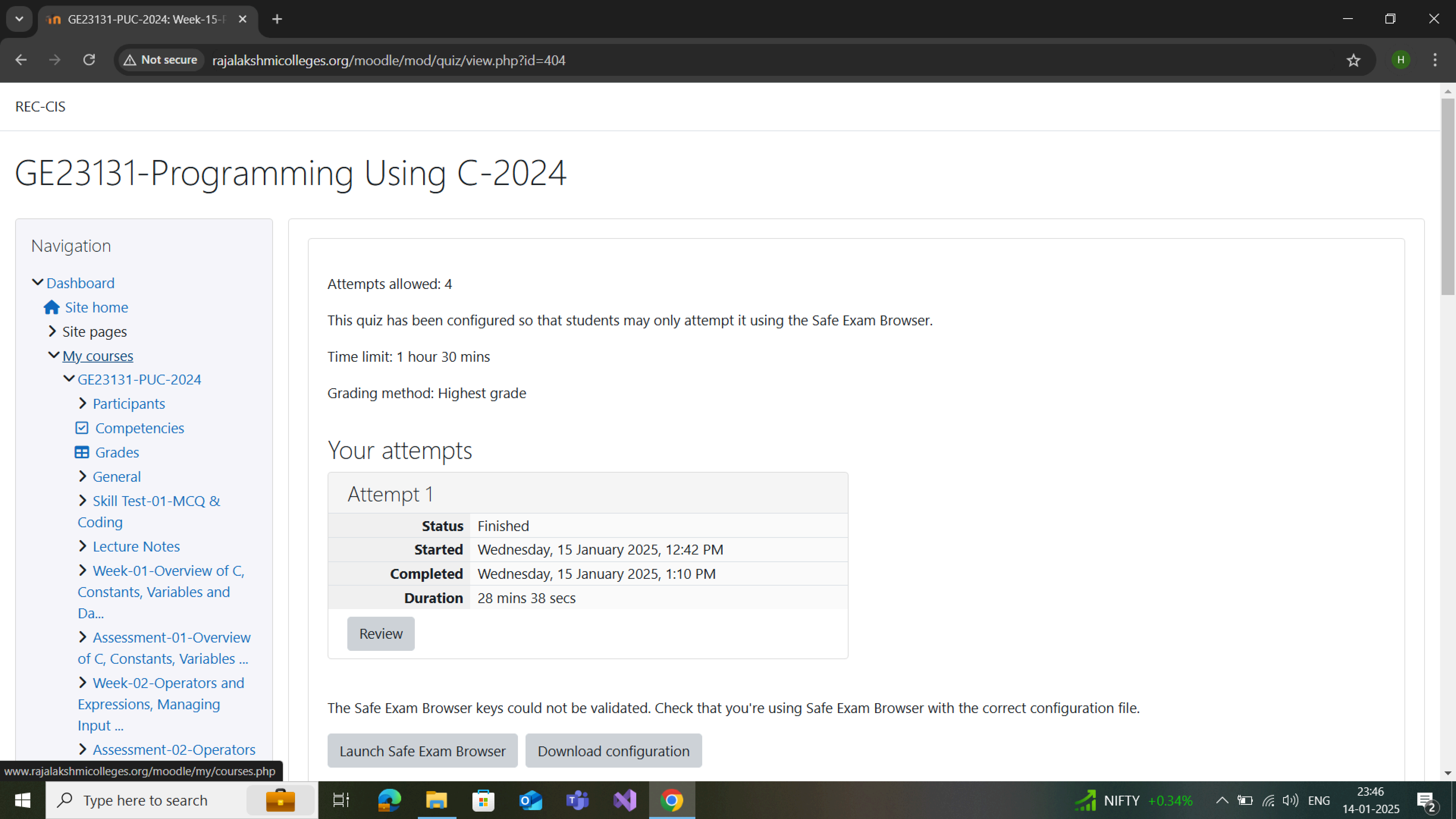
Attempt 1

Status	Finished
Started	Wednesday, 15 January 2025, 12:42 PM
Completed	Wednesday, 15 January 2025, 1:10 PM
Duration	28 mins 38 secs

Review

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

Launch Safe Exam Browser   Download configuration



GE23131-PUC-2024: Week-15-

rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=404

REC-CIS

GE23131-Programming Using C-2024

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - GE23131-PUC-2024
      - Participants
      - Competencies
      - Grades
      - General
      - Skill Test-01-MCQ & Coding
      - Lecture Notes
      - Week-01-Overview of C, Constants, Variables and Da...
      - Assessment-01-Overview of C, Constants, Variables ...
      - Week-02-Operators and Expressions, Managing Input ...
      - Assessment-02-Operators

Attempts allowed: 4

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 1 hour 30 mins

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Wednesday, 15 January 2025, 12:42 PM
Completed	Wednesday, 15 January 2025, 1:10 PM
Duration	28 mins 38 secs

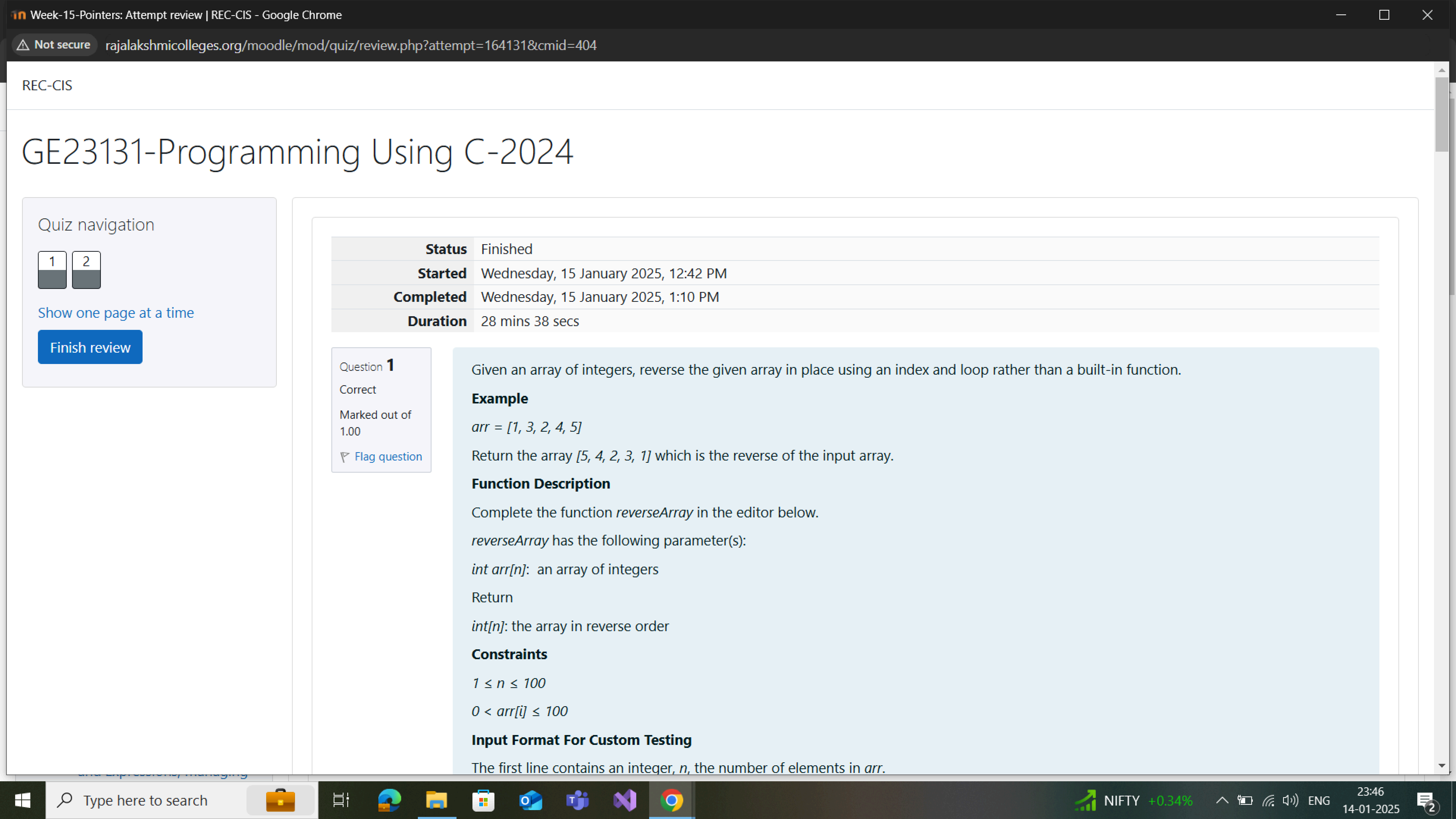
Review

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

Launch Safe Exam Browser Download configuration

www.rajalakshmicolleges.org/moodle/my/courses.php

Type here to search NIFTY +0.34% ENG 23:46 14-01-2025


Week-15-Pointers: Attempt review | REC-CIS - Google Chrome
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404
REC-CIS

GE23131-Programming Using C-2024

Quiz navigation
1
2
Show one page at a time
Finish review

Status	Finished
Started	Wednesday, 15 January 2025, 12:42 PM
Completed	Wednesday, 15 January 2025, 1:10 PM
Duration	28 mins 38 secs

Question **1**

Correct

Marked out of 1.00

Flag question

Given an array of integers, reverse the given array in place using an index and loop rather than a built-in function.

Example

arr = [1, 3, 2, 4, 5]

Return the array *[5, 4, 2, 3, 1]* which is the reverse of the input array.

Function Description

Complete the function *reverseArray* in the editor below.

reverseArray has the following parameter(s):

int arr[n]: an array of integers

Return

int[n]: the array in reverse order

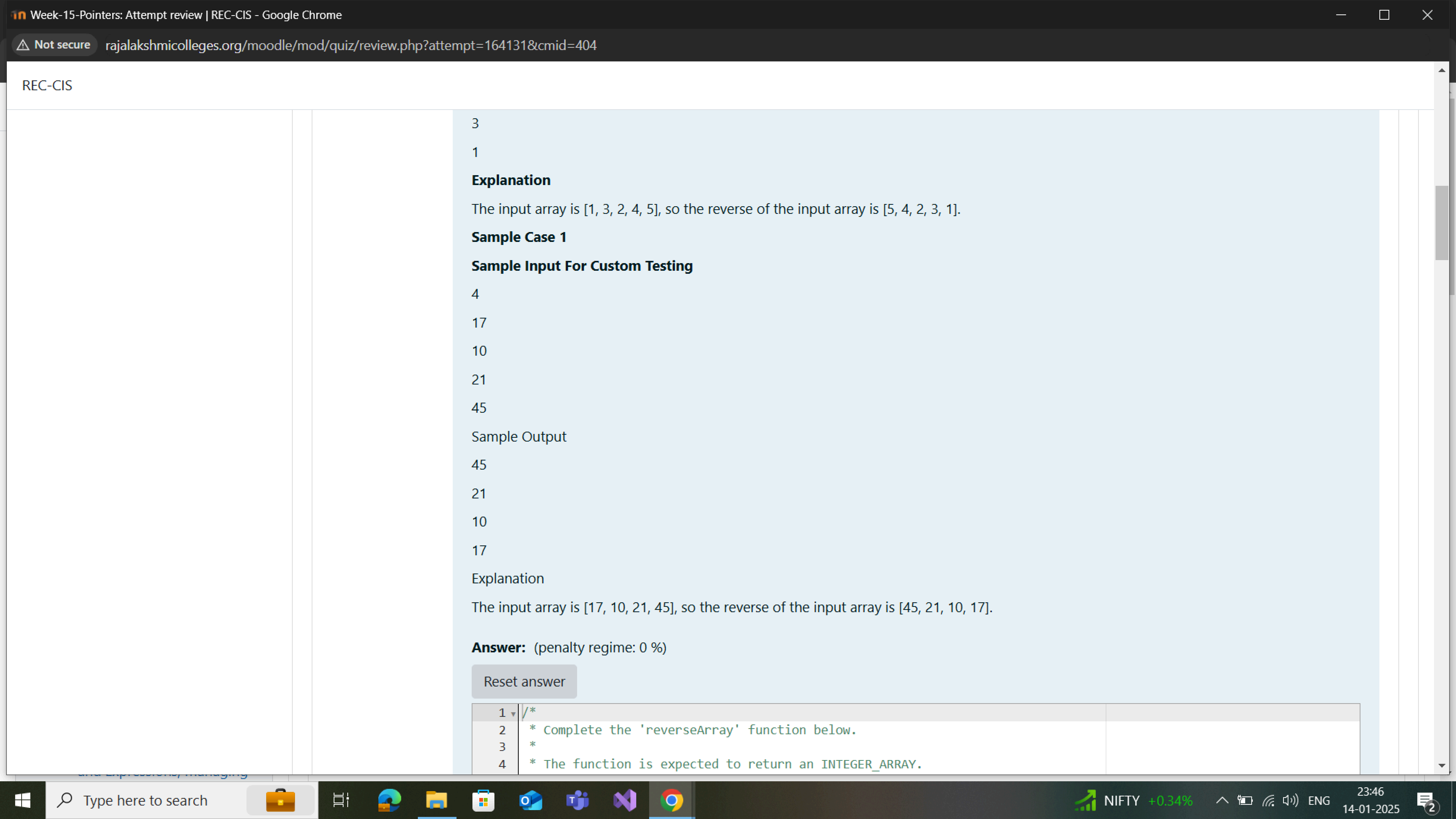
Constraints

$1 \le n \le 100$

$0 < arr[i] \le 100$

Input Format For Custom Testing

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


Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

3

1

Explanation

The input array is [1, 3, 2, 4, 5], so the reverse of the input array is [5, 4, 2, 3, 1].

Sample Case 1

Sample Input For Custom Testing

4

17

10

21

45

Sample Output

45

21

10

17

Explanation

The input array is [17, 10, 21, 45], so the reverse of the input array is [45, 21, 10, 17].

Answer: (penalty regime: 0 %)

Reset answer

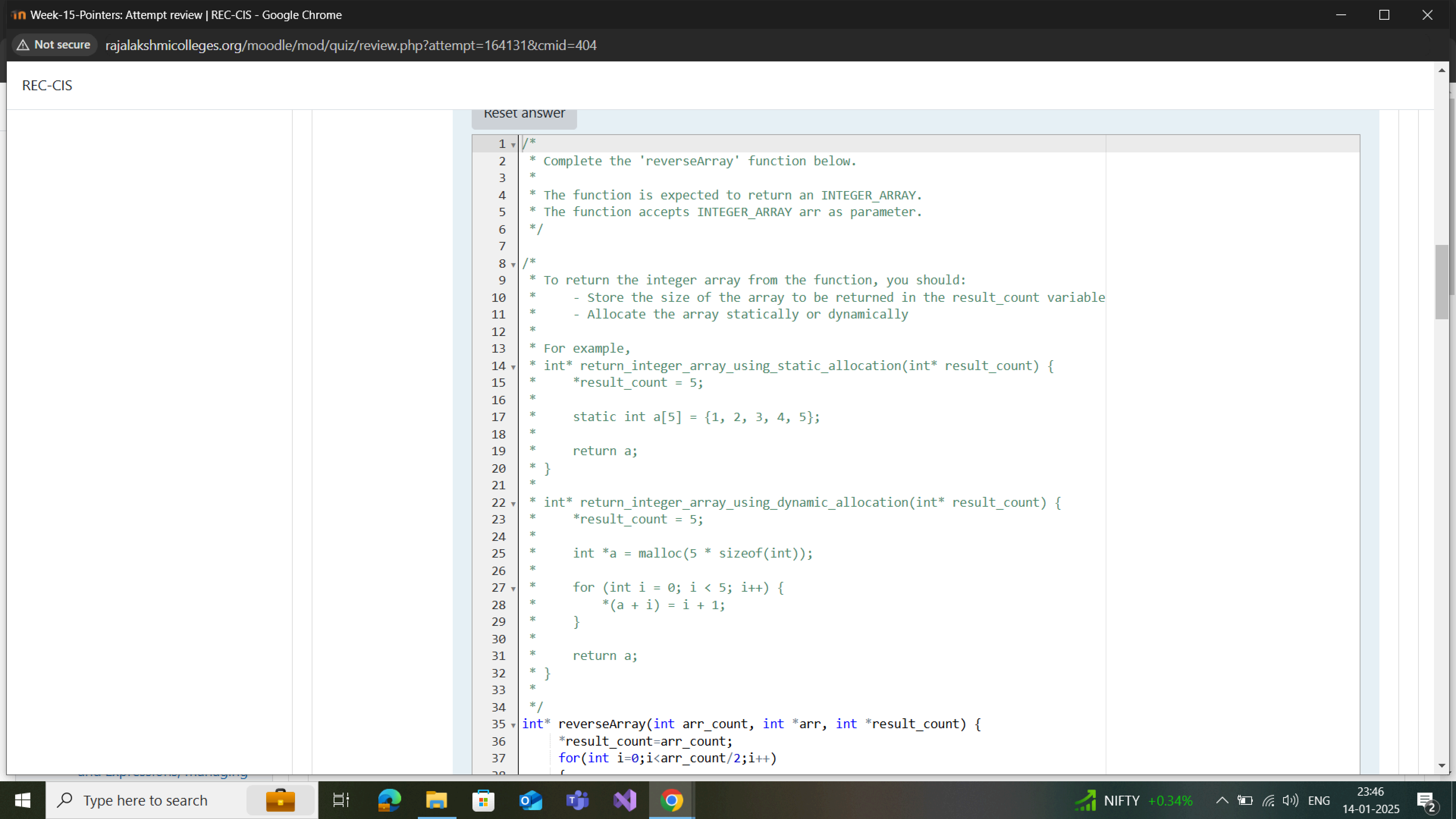
```
/*
 * Complete the 'reverseArray' function below.
 *
 * The function is expected to return an INTEGER_ARRAY.
```

Type here to search

NIFTY +0.34%

ENG

23:46
14-01-2025



Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

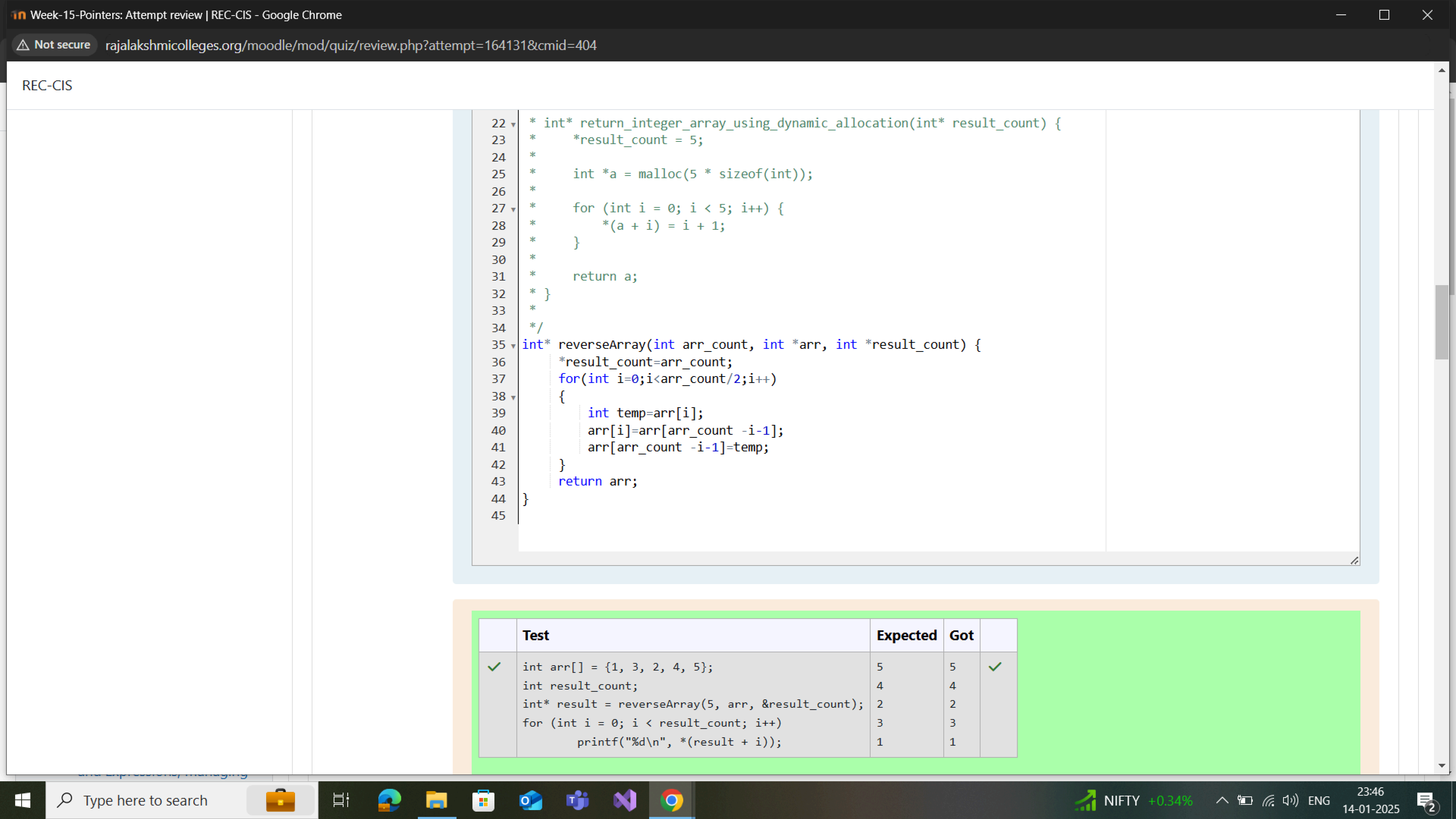
REC-CIS

Reset answer

```c
/*
 * Complete the 'reverseArray' function below.
 *
 * The function is expected to return an INTEGER_ARRAY.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

/*
 * To return the integer array from the function, you should:
 *     - Store the size of the array to be returned in the result_count variable
 *     - Allocate the array statically or dynamically
 *
 * For example,
 * int* return_integer_array_using_static_allocation(int* result_count) {
 *     *result_count = 5;
 *
 *     static int a[5] = {1, 2, 3, 4, 5};
 *
 *     return a;
 * }
 *
 * int* return_integer_array_using_dynamic_allocation(int* result_count) {
 *     *result_count = 5;
 *
 *     int *a = malloc(5 * sizeof(int));
 *
 *     for (int i = 0; i < 5; i++) {
 *         *(a + i) = i + 1;
 *     }
 *
 *     return a;
 * }
 *
 */
int* reverseArray(int arr_count, int *arr, int *result_count) {
    *result_count=arr_count;
    for(int i=0;i<arr_count/2;i++)
```

Type here to search

NIFTY +0.34% ENG 23:46 14-01-2025



Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

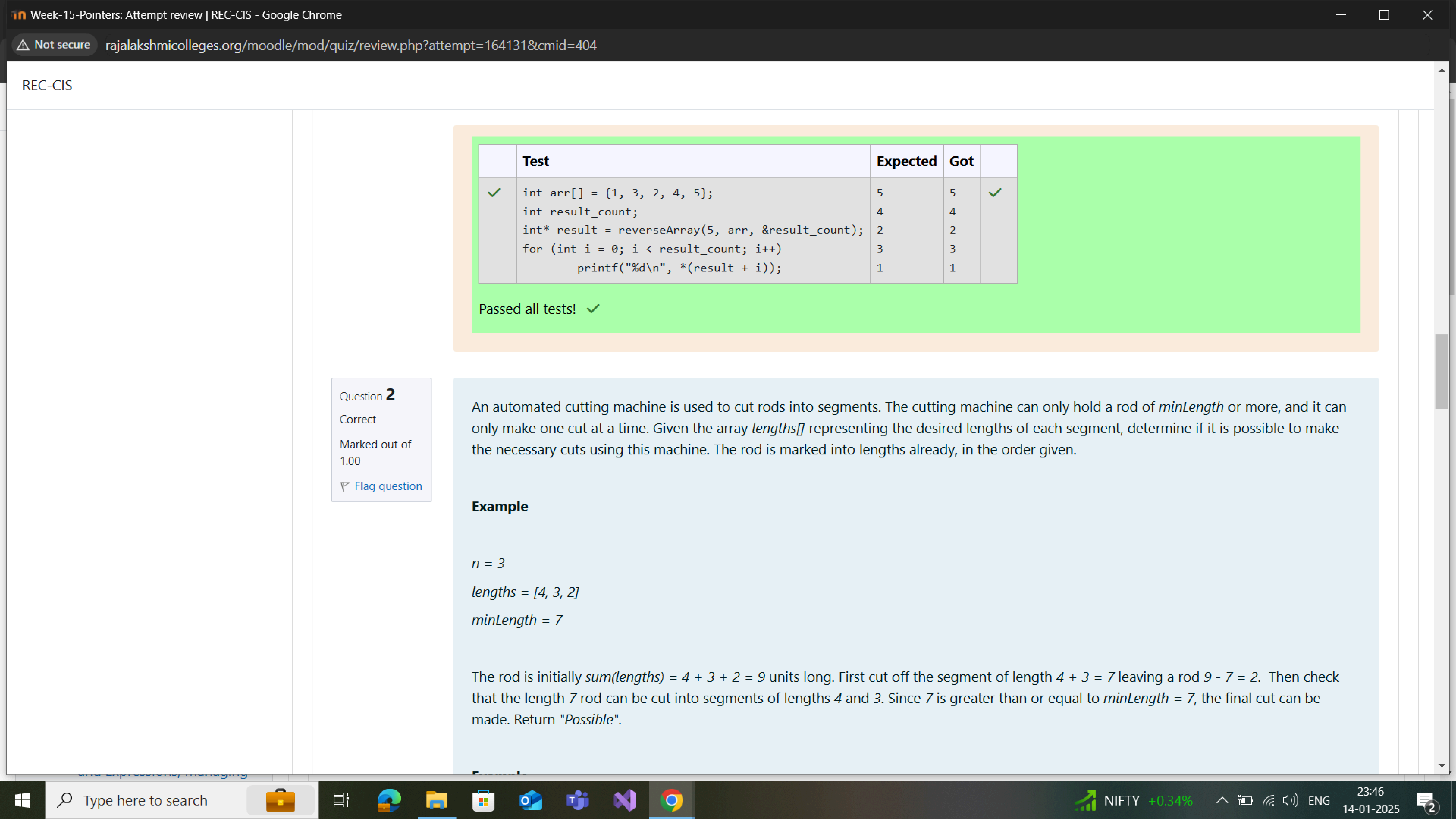
```
 * int* return_integer_array_using_dynamic_allocation(int* result_count) {
 *     *result_count = 5;
 *
 *     int *a = malloc(5 * sizeof(int));
 *
 *     for (int i = 0; i < 5; i++) {
 *         *(a + i) = i + 1;
 *     }
 *
 *     return a;
 * }
 *
 */
int* reverseArray(int arr_count, int *arr, int *result_count) {
    *result_count=arr_count;
    for(int i=0;i<arr_count/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[arr_count -i-1];
        arr[arr_count -i-1]=temp;
    }
    return arr;
}

```

	Test	Expected	Got	

✓	`int arr[] = {1, 3, 2, 4, 5};`
	`int result_count;`
	`int* result = reverseArray(5, arr, &result_count);`
	`for (int i = 0; i < result_count; i++)`
	`        printf("%d\n", *(result + i));`
		5
		4
		2
		3
		1
			5
			4
			2
			3
			1
				✓

Type here to search

NIFTY +0.34% ENG 23:46 14-01-2025



Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

	Test	Expected	Got	

✓	`int arr[] = {1, 3, 2, 4, 5};`
	`int result_count;`
	`int* result = reverseArray(5, arr, &result_count);`
	`for (int i = 0; i < result_count; i++)`
	`        printf("%d\n", *(result + i));`
		5
		4
		2
		3
		1
			5
			4
			2
			3
			1
				✓

Passed all tests! ✓

Question **2**
Correct
Marked out of 1.00
Flag question

An automated cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of *minLength* or more, and it can only make one cut at a time. Given the array *lengths[]* representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using this machine. The rod is marked into lengths already, in the order given.

Example

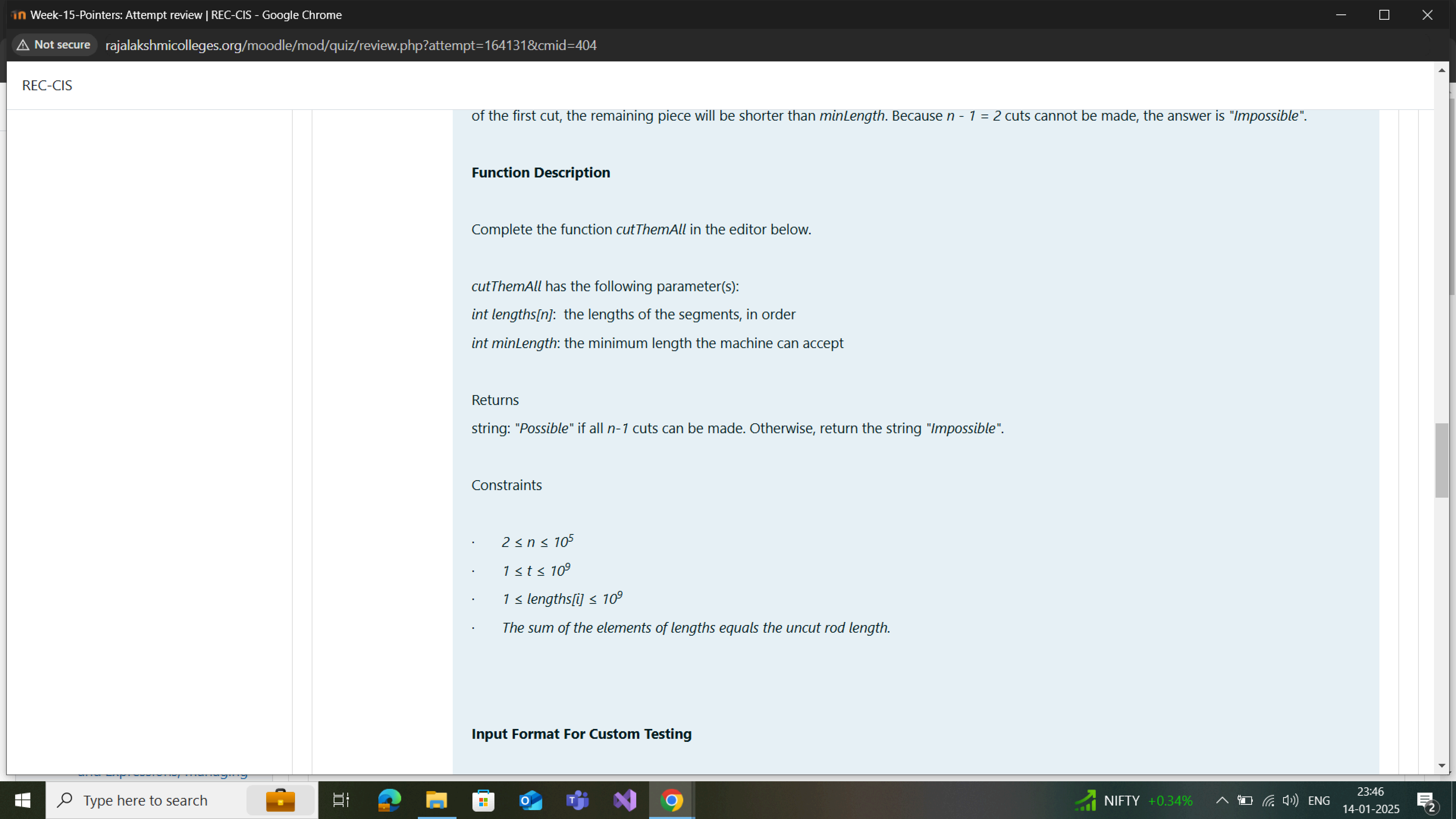
n = 3

lengths = [4, 3, 2]

minLength = 7

The rod is initially *sum(lengths) = 4 + 3 + 2 = 9* units long. First cut off the segment of length *4 + 3 = 7* leaving a rod *9 - 7 = 2*. Then check that the length *7* rod can be cut into segments of lengths *4* and *3*. Since *7* is greater than or equal to *minLength = 7*, the final cut can be made. Return "*Possible*".

Type here to search   NIFTY +0.34%   ENG   23:46 14-01-2025



Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

of the first cut, the remaining piece will be shorter than *minLength*. Because *n - 1 = 2* cuts cannot be made, the answer is "*Impossible*".

Function Description

Complete the function *cutThemAll* in the editor below.

cutThemAll has the following parameter(s):

int lengths[n]: the lengths of the segments, in order

int minLength: the minimum length the machine can accept

Returns

string: "*Possible*" if all *n-1* cuts can be made. Otherwise, return the string "*Impossible*".

Constraints

- $2 \le n \le 10^5$
- $1 \le t \le 10^9$
- $1 \le lengths[i] \le 10^9$
- *The sum of the elements of lengths equals the uncut rod length.*

Input Format For Custom Testing

Type here to search

NIFTY +0.34%

ENG

23:46

14-01-2025

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

Each line *i* of the *n* subsequent lines (where $0 \le i < n$) contains an integer, *lengths[i]*.

The next line contains an integer, *minLength*, the minimum length accepted by the machine.

Sample Case 0

Sample Input For Custom Testing

```
STDIN     Function
-----    --------
4    →  lengths[] size n = 4
3    →  lengths[] =  [3, 5, 4, 3]
5
4
3
9    →  minLength= 9
```

Sample Output

```
Possible
```

Explanation

Sample Case 1

Sample Input For Custom Testing

```
STDIN    Function
-----    --------
3    →  lengths[] size n = 3
5    →  lengths[] =  [5, 6, 2]
6
2
12   →  minLength= 12
```

Sample Output

```
Impossible
```

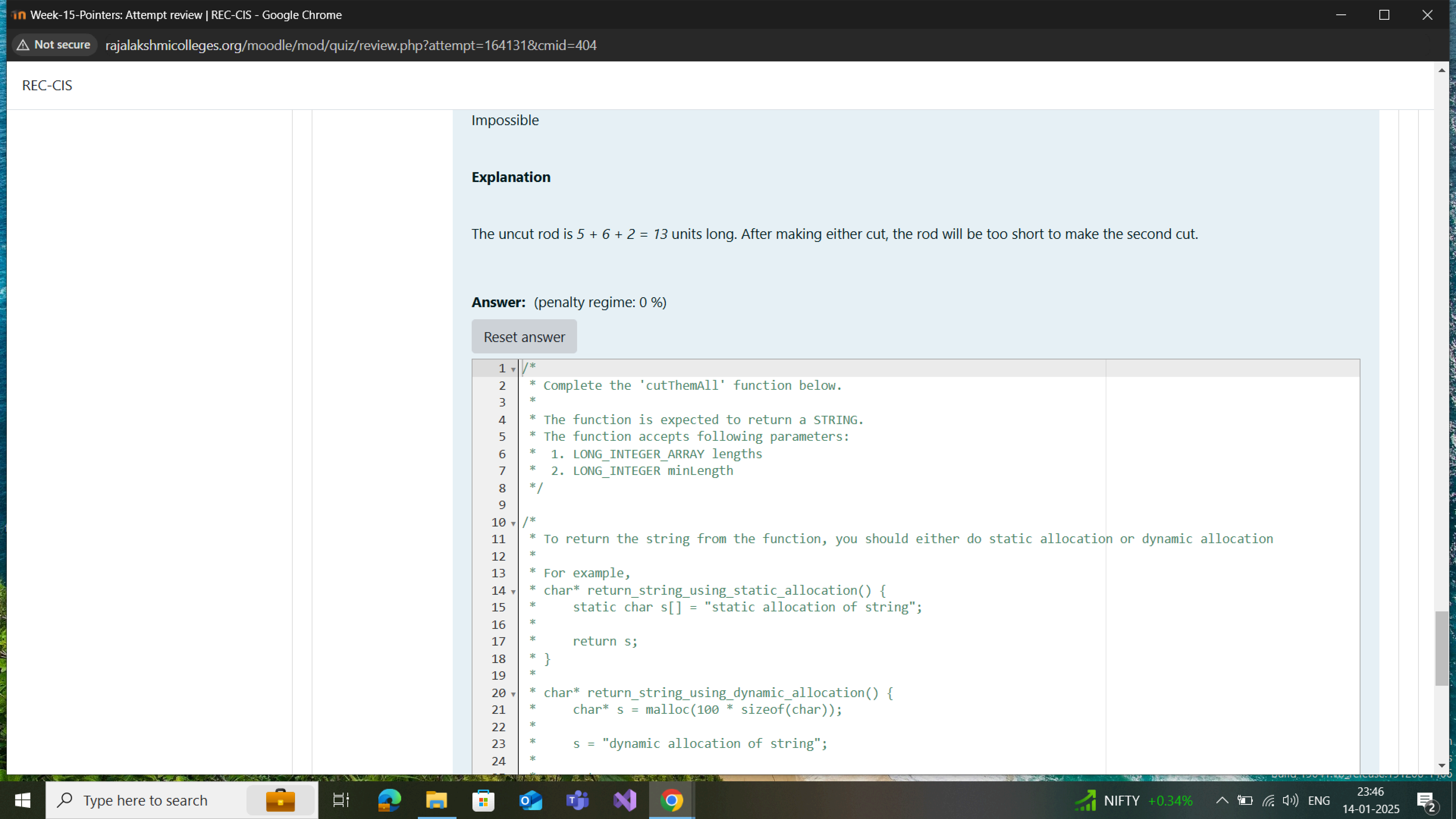
Explanation

The uncut rod is *5 + 6 + 2 = 13* units long. After making either cut, the rod will be too short to make the second cut.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'cutThemAll' function below.
```



Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

Impossible

Explanation

The uncut rod is *5 + 6 + 2 = 13* units long. After making either cut, the rod will be too short to make the second cut.

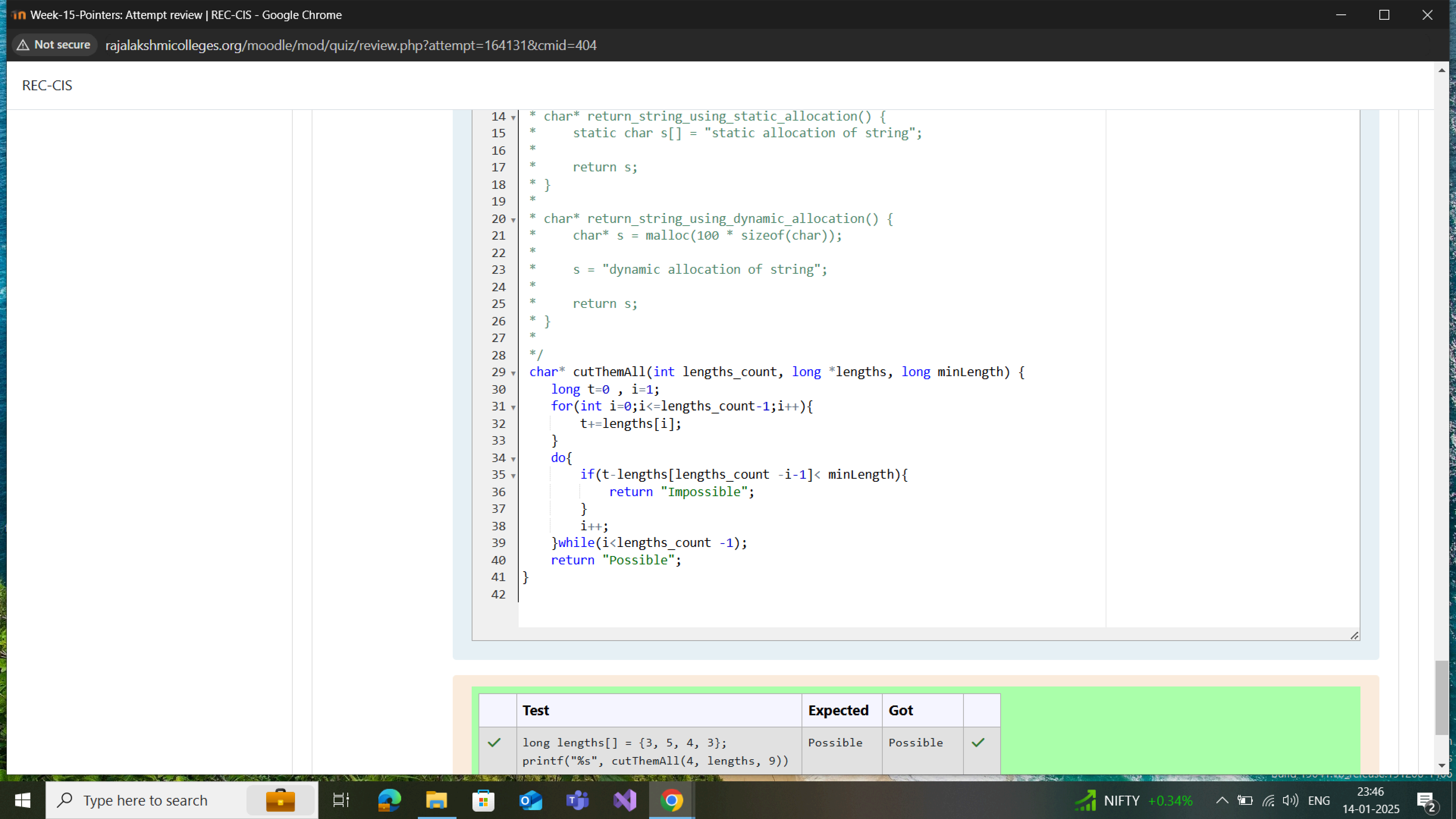
Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'cutThemAll' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts following parameters:
 *  1. LONG_INTEGER_ARRAY lengths
 *  2. LONG_INTEGER minLength
 */

/*
 * To return the string from the function, you should either do static allocation or dynamic allocation
 *
 * For example,
 * char* return_string_using_static_allocation() {
 *     static char s[] = "static allocation of string";
 *
 *     return s;
 * }
 *
 * char* return_string_using_dynamic_allocation() {
 *     char* s = malloc(100 * sizeof(char));
 *
 *     s = "dynamic allocation of string";
 *
```

Type here to search

NIFTY +0.34% ENG 23:46 14-01-2025


Week-15-Pointers: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404

REC-CIS

```c
 * char* return_string_using_static_allocation() {
 *     static char s[] = "static allocation of string";
 *
 *     return s;
 * }
 *
 * char* return_string_using_dynamic_allocation() {
 *     char* s = malloc(100 * sizeof(char));
 *
 *     s = "dynamic allocation of string";
 *
 *     return s;
 * }
 *
 */
char* cutThemAll(int lengths_count, long *lengths, long minLength) {
    long t=0 , i=1;
    for(int i=0;i<=lengths_count-1;i++){
        t+=lengths[i];
    }
    do{
        if(t-lengths[lengths_count -i-1]< minLength){
            return "Impossible";
        }
        i++;
    }while(i<lengths_count -1);
    return "Possible";
}
```

	Test	Expected	Got	

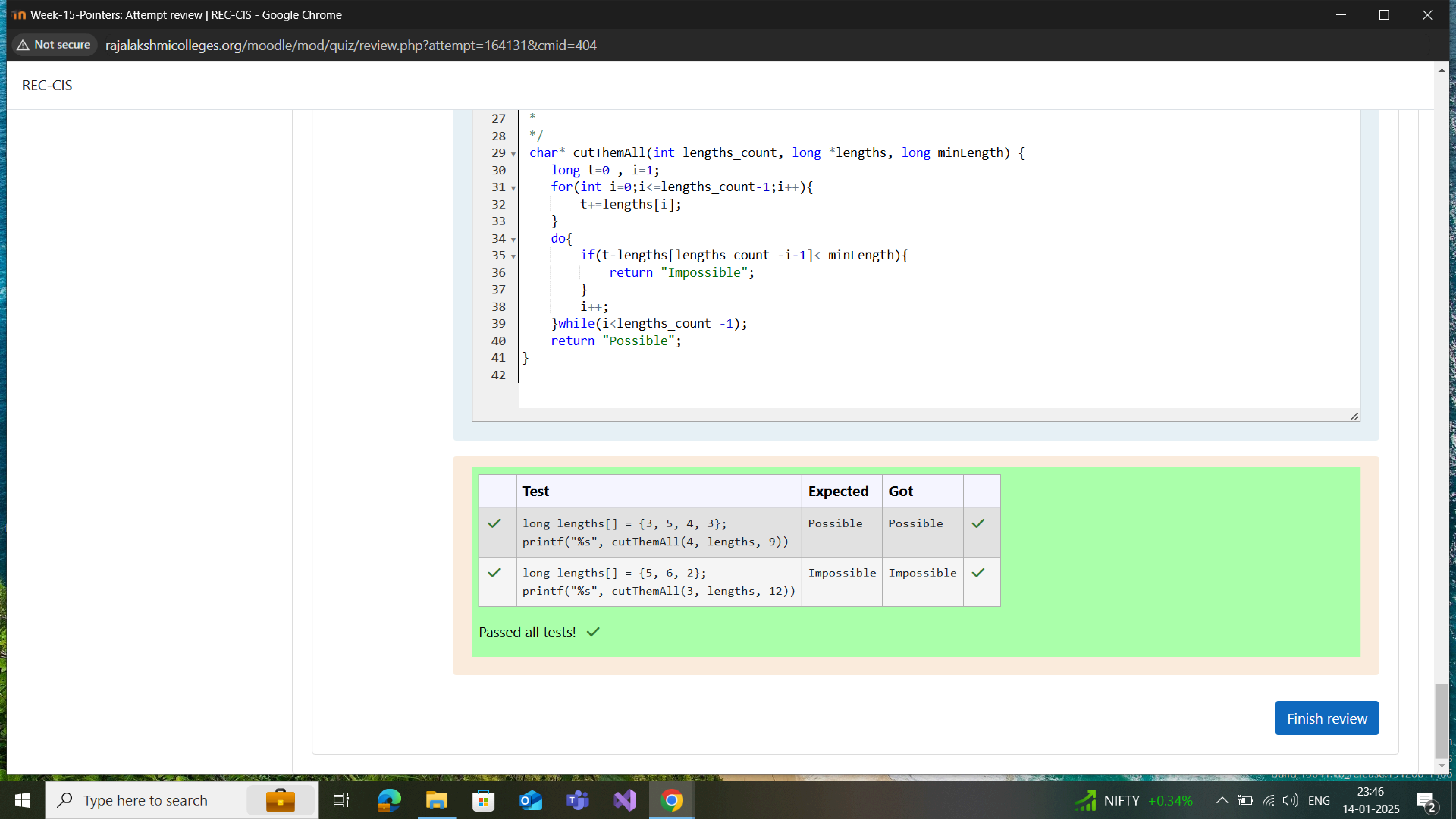
✓	long lengths[] = {3, 5, 4, 3};
	printf("%s", cutThemAll(4, lengths, 9))
		Possible	Possible	✓

Type here to search

NIFTY +0.34%

ENG

23:46
14-01-2025


Week-15-Pointers: Attempt review | REC-CIS - Google Chrome
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=164131&cmid=404
REC-CIS

```
 *
 */
char* cutThemAll(int lengths_count, long *lengths, long minLength) {
    long t=0 , i=1;
    for(int i=0;i<=lengths_count-1;i++){
        t+=lengths[i];
    }
    do{
        if(t-lengths[lengths_count -i-1]< minLength){
            return "Impossible";
        }
        i++;
    }while(i<lengths_count -1);
    return "Possible";
}
```

	Test	Expected	Got	
✓	`long lengths[] = {3, 5, 4, 3};` `printf("%s", cutThemAll(4, lengths, 9))`	Possible	Possible	✓
✓	`long lengths[] = {5, 6, 2};` `printf("%s", cutThemAll(3, lengths, 12))`	Impossible	Impossible	✓

Passed all tests! ✓