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

REC-CIS

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
      - 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 ...
      - Week-03-Decision Making and Branching - if, if...e...
      - Assessment-03-Decision Making and Branching - if, ...

◄Week-14-Structures and Unions

Week-15-Pointers

◄Week-14-Structures and Unions

Jump to...

✓ Done

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	Friday, 17 January 2025, 2:03 AM
Completed	Friday, 17 January 2025, 2:09 AM
Duration	6 mins 17 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

Back to the course

```
28    *      (a + 1) = 1 + 1;
29    *    }
30    *
31    *        return a;
32    * }
33    *
34    */
35    #include<stdio.h>
36    #include<stdlib.h>
37 ▾ int* reverseArray(int arr_count, int *arr, int *result_count) {
38        int* result=(int*)malloc(arr_count*sizeof(int));
39        if(result==NULL)
40 ▾    {
41            return NULL;
42        }
43        for(int i=0;i<arr_count;i++)
44 ▾    {
45            result[i]=arr[arr_count-i-1];
46        }
47        *result_count=arr_count;
48        return result;
49
50 }
51
```

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! ✓

```
22   *
23   *        s = "dynamic allocation of string";
24   *
25   *        return s;
26   * }
27   *
28   */
29 ▾ char* cutThemAll(int lengths_count, long *lengths, long minLength) {
30        long t=0, i=1;
31        for(int i=0;i<=lengths_count-1;i++)
32 ▾     {
33            t+=lengths[i];
34
35        }
36 ▾     do{
37            if(t-lengths[lengths_count-1]<minLength)
38 ▾        {
39                return "Impossible";
40            }
41            i++;
42
43        }while(i<lengths_count-1);
44        return "Possible";
45
46 }
47
```

	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! ✓