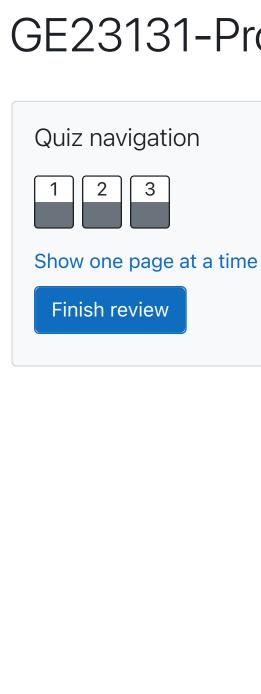
**REC-CIS** 

## GE23131-Programming Using C-2024



Status Finished **Duration** 66 days 6 hours Question 1 Correct Marked out of 3.00 question

**Objective** 

**Output Format** 

**Sample Output** 

Hello, World!

2 3 •

4 5

Print *Hello, World!* to stdout.

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

int main()

**Expected** 

Passed all tests! <

sentence as input in C.

**Objective** 

stdout:

char ch;

Task

**Input Format** 

**Output Format** 

Print the character, *ch*.

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

1 #include <stdio.h>

Input Expected Got

is scanf("format string",argument\_list);. For ex:

and stores the given value in variable *number*.

type as input and output their sum:

place on a new line.

**Input Format** 

**Output Format** 

10 4

14 6

6.0 2.0

difference.

2

4

5

7

8

Passed all tests! <

3 ▼ {

Answer: (penalty regime: 0 %)

int main()

1 #include <stdio.h>

int a,b;

float c,d;

scanf("%d %d",&a,&b); scanf("%f %f", &c,&d);

printf("%d %d\n",a+b,a-b);

4.0 2.0

**Sample Output** 

Task

Question **3** 

Marked out of

Correct

7.00

▼ Flag question Take a character, **ch** as input.

scanf("%c", &ch);

printf("%c", ch);

Question 2

Marked out of

Correct

5.00

▼ Flag question Hello, World!

#include <stdio.h>

printf("Hello, World!");

Got

Hello, World!

This challenge will help you to learn how to take a character, a string and a

&ch); and printf("%c", ch) writes a character specified by the argument char to

To take a single character ch as input, you can use scanf("%c",

**/** 

**Input Format** 

World! to stdout.

Started Monday, 23 December 2024, 5:33 PM Completed Friday, 18 October 2024, 11:28 AM This is a simple challenge to help you practice printing to stdout.

We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string Hello, You do not need to read any input in this challenge.

This piece of code prints the character *ch*. You have to print the character, ch.

int main() 2 3 ▼ { char ch; 4 scanf("%c",&ch); 5 printf("%c",ch); 6 7

C C **✓** Passed all tests! < **Objective** The fundamental data types in c are int, float and char. Today, we're discussing int and float data types. The printf() function prints the given statement to the console. The syntax is printf("format string", argument\_list);. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively. The scanf() function reads the input data from the console. The syntax

The scanf("%d",&number) statement reads integer number from the console

To input two integers separated by a space on a single line, the command

Your task is to take two numbers of int data type, two numbers of float data

is scanf("%d %d", &n, &m), where n and m are the two integers.

Declare 4 variables: two of type int and two of type float. Read 2 lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your 4 variables. Use the + and - operator to perform the following operations: Print the sum and difference of two int variable on a new line.

o Print the sum and difference of two float variable rounded to one decimal

Print the sum and difference of both integers separated by a space on the first

line, and the sum and difference of both float (scaled to 1 decimal place)

## **Constraints** 1 ≤ integer variables ≤ 10<sup>4</sup> 1 ≤ float variables ≤ 10<sup>4</sup>

The first line contains two integers.

The second line contains two floating point numbers.

separated by a space on the second line. **Sample Input** 

**Explanation** When we sum the integers 10 and 4, we get the integer 14. When we subtract the second number 4 from the first number 10, we get 6 as their difference.

When we sum the floating-point numbers 4.0 and 2.0, we get 6.0. When we

subtract the second number 2.0 from the first number 4.0, we get 2.0 as their

printf("%.1f %.1f",c+d,c-d); 9 10 return 0; 11 **Expected Got** Input 14 6 14 6 **✓** 10 4 4.0 2.0 6.0 2.0 6.0 2.0 28 12 20 8 28 12 8.0 4.0 | 12.0 4.0 12.0 4.0

Finish review