

You do not need to read any input in this challenge.
Output Format
Print <i>Hello, World!</i> to stdout.
Sample Output
Hello, World! Answer: (penalty regime: 0 %)
<pre>#include<stdio.h> int main(){ printf("Hello, World!"); return 0; }</stdio.h></pre>

	Expected	Got	
~	Hello, World!	Hello, World!	~

Passed all tests! ✓

Objective

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character **ch** as input, you can use scanf("%c", &ch); and printf("%c", ch) writes a character specified by the argument char to stdout:

char ch; scanf("%c", &ch); printf("%c", ch);

This piece of code prints the character <i>ch</i> .	
Task	
You have to print the character, <i>ch</i> .	
Input Format	
Take a character, <i>ch</i> as input.	
Output Format	
Print the character, ch .	
Answer: (penalty regime: 0 %)	

```
3
       char ch;
       scanf("%c",&ch);
4
       printf("%c",ch);
5
6
       return 0;
```

Question 3 Correct	Objective
Marked out of 7.00 ▼ Flag question	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 is scanf("format string", argument_list);. For ex: The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable <i>number</i> .
	To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where $\bf n$ and $\bf m$ are the two integers.
	Task
	Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

- Declare 4 variables: two of type int and two of type float.
- 2. 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:
- o 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 place on a new line.

Input Format

The first line contains two integers.

The second line contains two floating point numbers.

Constraints

- 1 ≤ integer variables ≤ 10⁴
- 1 ≤ float variables ≤ 10⁴

Output Format

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) separated by a space on the second line.

Sample Input

4.0 2.0

104

146

6.0 2.0

Sample Output

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

scanf("\n%f %f",&C,&D);
printf("%d %d",A+B,A-B);
printf("\n%.1f %.1f",C+D,C-D);

return 0;

10 11 }

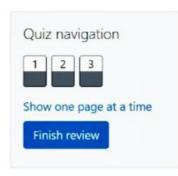
number **2.0** from the first number **4.0**, we get **2.0** as their difference.

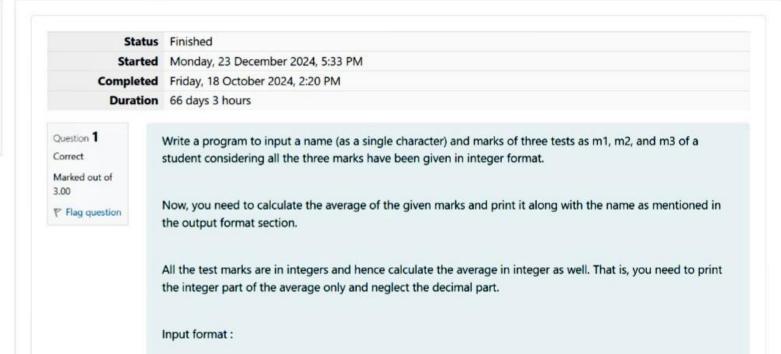
when we sum the integers 10 and 4, we get the integer 14. When we subtract the second number 4 from the



Finish review

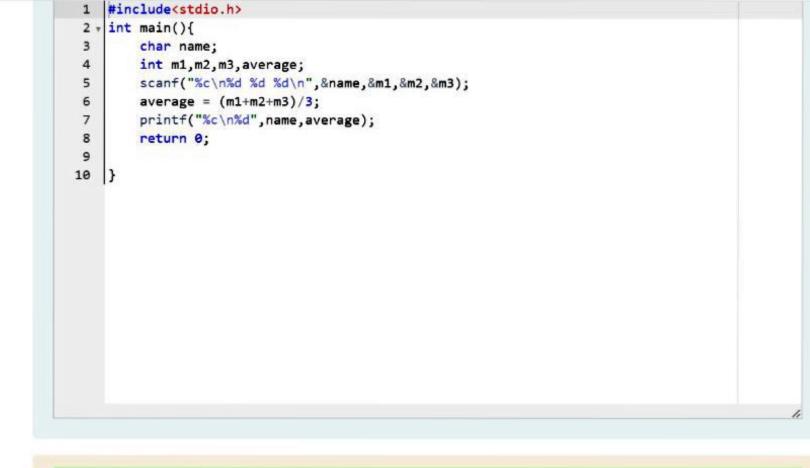
GE23131-Programming Using C-2024



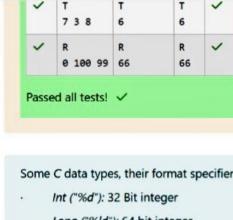


Input format :
Line 1 : Name(Single character) Line 2 : Marks scored in the 3 tests separated by single space.
Output format :
First line of output prints the name of the student.
Second line of the output prints the average mark. Constraints
Marks for each student lie in the range 0 to 100 (both inclusive)
Sample Input 1 :
A
346

	Sample Output 1 :	
	A	
	4	
	Sample Input 2:	
	Т	
	738	
	Sample Output 2 :	
	Т	
	6	
	Answer: (penalty regime: 0 %)	
	1 #include <stdio.h></stdio.h>	







Correct Marked out of 5.00 P Flag question

Question 2

Some C data types, their format specifiers, and their most common bit widths are as follows: Long ("%ld"): 64 bit integer

Char ("%c"): Character type Float ("%f"): 32 bit real value Double ("%lf"): 64 bit real value

char ch;

Reading

For example, to read a character followed by a double:

To read a data type, use the following syntax:

scanf("`format_specifier`", &val)

For the moment, we can ignore the spacing between format specifiers.

Printing

double d:

scanf("%c %lf", &ch, &d);

To print a data type, use the following syntax: printf("`format_specifier`", val)

For example, to print a character followed by a double:

printf("%c %lf", ch, d);

double d = 234.432;

char ch = 'd';

Note: You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

Input Format

Input consists of the following space-separated values: int, long, char, float, and double, respectively.

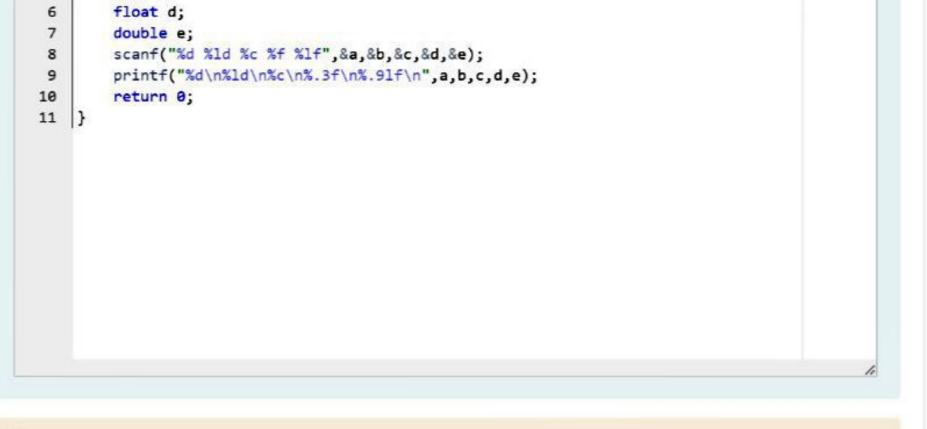
Output Format

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

2 12245670012245 2 224 22 14040 20402

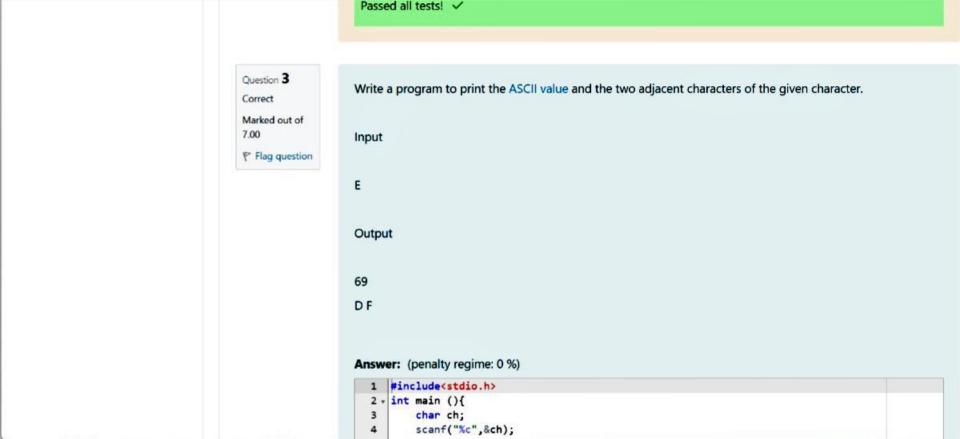
```
should be correct up to 3 decimal places and the double to 9 decimal places.
Sample Input
3 12345678912345 a 334.23 14049.30493
Sample Output
12345678912345
a
334.230
14049.304930000
Explanation
Print int 3,
followed by long 12345678912345,
followed by char a,
followed by float 334.23,
followed by double 14049.30493.
Answer: (penalty regime: 0 %)
   1 #include<stdio.h>
   2 - int main (){
           int a;
```

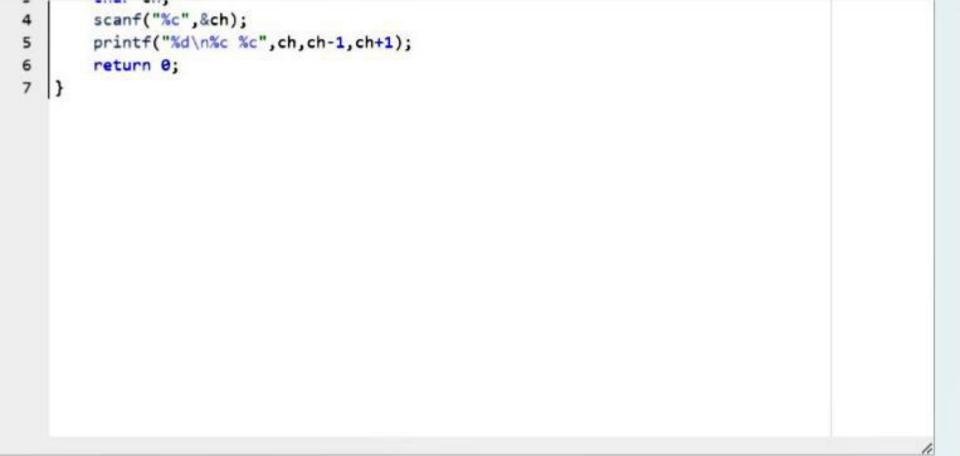


	Input	Expected	Got	
~	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a	3 12345678912345 a	~
		334.230	334.230	
		14049.304930000	14049.304930000	

5

char c;





	Input	Expected	Got	
~	E	69	69	~
	100	DF	DF	

Passed all tests! <