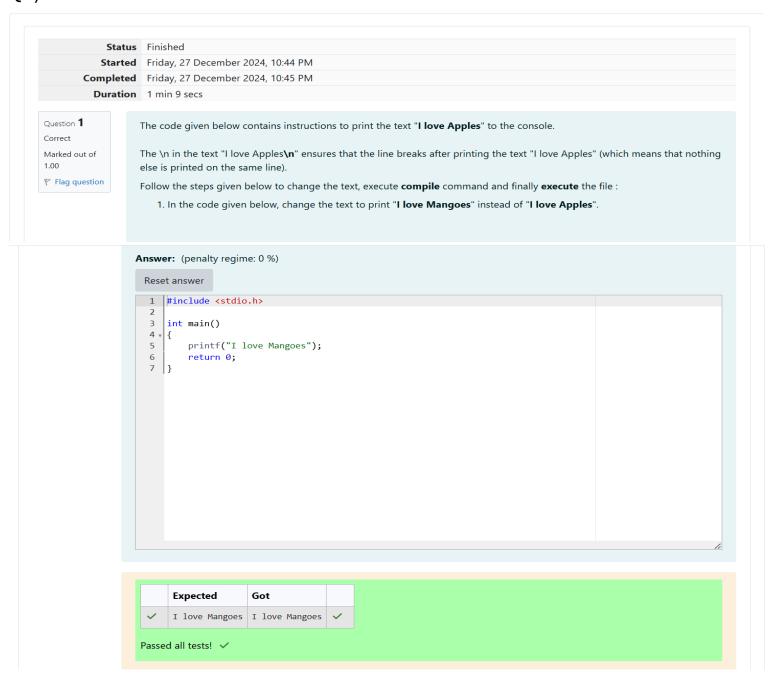
# Week-01-Overview of C, Constants, Variables and Data Types

--Coding-C-Language Features-Optional.

ROLL NO: 241801294

NAME: THARUN N

Q1)

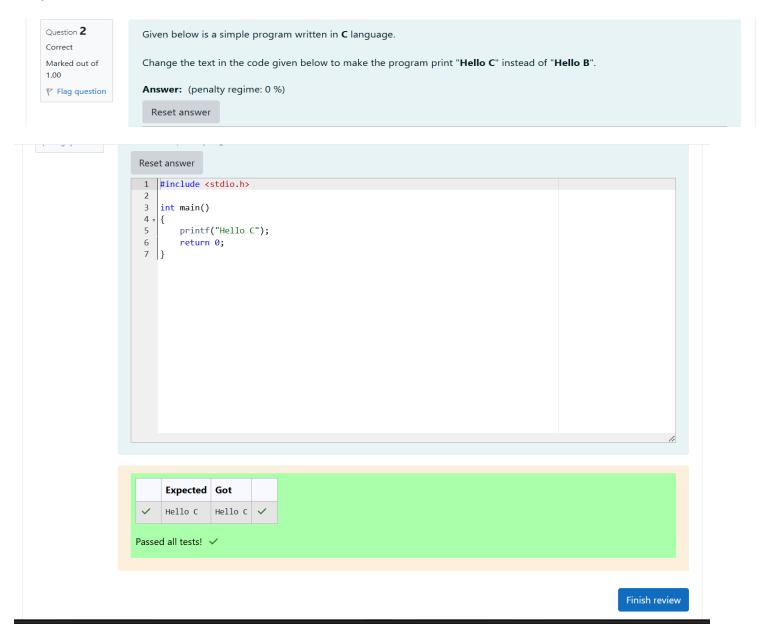


--Coding-C-Language Features-Optional.

ROLL NO: 241801294

NAME: THARUN N

Q2)

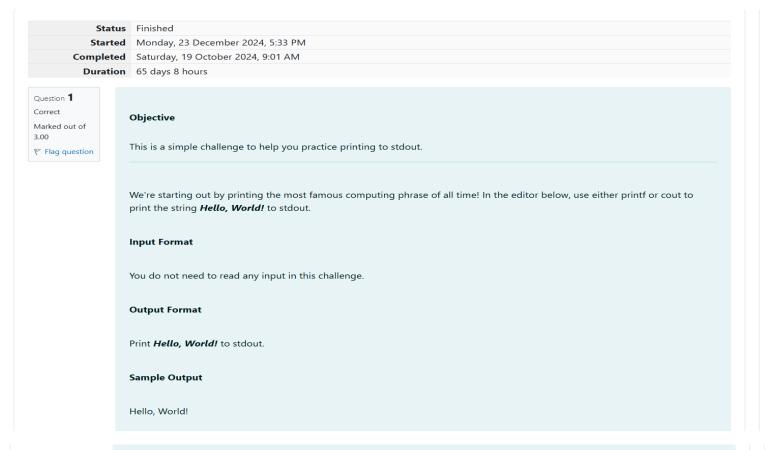


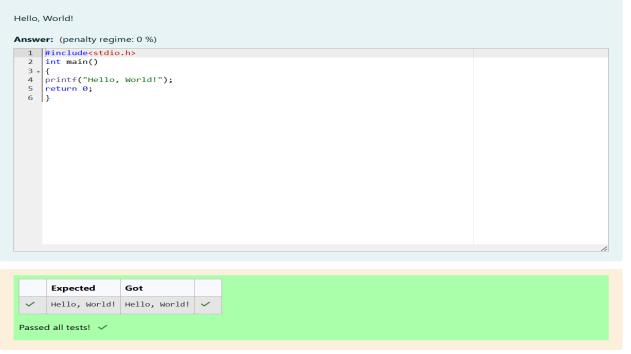
# Week-01-01-Practice Session-Coding

ROLL NO: 241801294

NAME: THARUN N

Q1)

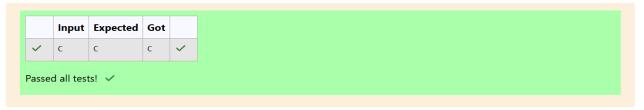






Print the character, ch.

# 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 ch. Task You have to print the character, ch. Input Format Take a character, ch as input.



Question **3**Correct
Marked out of 7.00

▼ Flag question

### 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 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 *number*.

To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where  $\mathbf{n}$  and  $\mathbf{m}$  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:

- 1. 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.
- 3. 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 ≤ 10<sup>4</sup>
- 1 ≤ float variables ≤ 10<sup>4</sup>

### **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

10 4

4.0 2.0

### **Sample Output**

14 6

6.0 2.0

### **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 difference.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
  2
         int main()
  3 ▼ {
  4
                 int a, b;
         int a, b;
    float c, d;
    scanf("%d", &a);
    scanf("%d", &b);
    scanf("%f", &c);
    scanf("%f", &d);
    printf("%d'n, (a+b));
    printf("%.1f", (c+d));
    printf(" %.1f", (c-d));
    return 0:
  5
 6
  7
  8
  9
10
11
12
13
                 return 0;
14
15 }
```

	Input	Expected	Got	
<b>~</b>	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	~
<b>~</b>		28 12 12.0 4.0	28 12 12.0 4.0	~

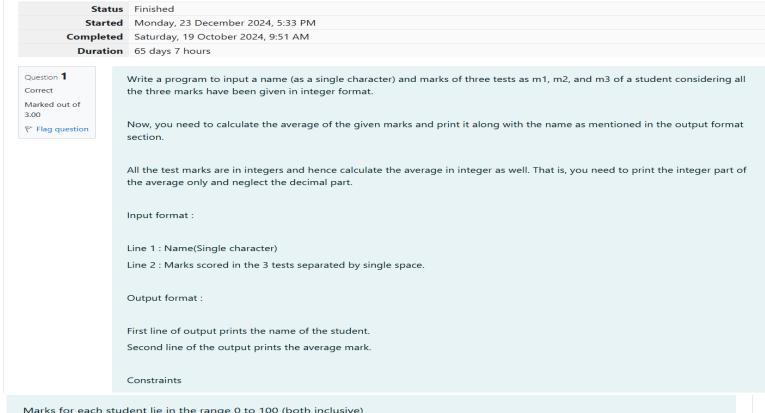
Finish review

## Week-01-02-Practice Session-Coding

ROLL NO: 241801294

NAME: THARUN N

Q1)



Marks for each student lie in the range 0 to 100 (both inclusive)
Sample Input 1 :
A
3 4 6
Sample Output 1 :
A
4
Sample Input 2 :
т
7 3 8
Sample Output 2 :
т
6
Answer: (penalty regime: 0 %)

```
Answer: (penalty regime: 0 %)
     1 #include<stdio.h>
      2 int main()
     3 ₹ {
      4
                   char ch;
              char ch;
int a, b, c;
scanf("%c", &ch);
scanf("%d", &a);
scanf("%d", &b);
scanf("%d", &c);
printf("%c\n", ch);
printf("%d", ((a+b+c)/3));
return 0:
      5
      6
      8
      9
    10
    11
    12
                  return 0;
    13 }
```

	Input	Expected	Got	
~	A 3 4 6	A 4	A 4	<b>~</b>
~	T 7 3 8	T 6	T 6	<b>&gt;</b>
~	R Ø 100 99	R 66	R 66	<b>~</b>

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Flag question

Some C data types, their format specifiers, and their most common bit widths are as follows:

- Int ("%d"): 32 Bit integer
- · Long ("%ld"): 64 bit integer
- · Char ("%c"): Character type
- · Float ("%f"): 32 bit real value
- · Double ("%lf"): 64 bit real value

### Reading

To read a data type, use the following syntax:

scanf("`format\_specifier`", &val)

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

char ch;

double d;

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

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

### Printing

To print a data type, use the following syntax:

printf("`format\_specifier`", val)

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

char ch = 'd';

double d = 234.432;

printf("%c %lf", 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

3 12345678912345 a 334.23 14049.30493

### **Sample Output**

3

12345678912345

а

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> int main() 3 🔻 { 4 int a; long b; char ch; float d; double e; scanf("%d", &a); scanf("%ld\n", &b); 5 6 scanf("%c\n", &ch); scanf("%f\n", &d); 7 8 scanf("%lf\n", &e); 9 printf("%d\n", a); printf("%ld\n", b); printf("%c\n", ch); printf("%.3f\n", d); printf("%.91f\n", e); 10 11 12 13 14 15 return 0; 16 }

	Input	Expected	Got	
<b>~</b>	3 12345678912345 a 334.23 14049.30493	3 12345678912345	3 12345678912345	<b>~</b>
		a 334.230 14049.304930000	a 334.230 14049.304930000	

Passed all tests! <

Question **3**Correct
Marked out of 7.00

Flag question

```
Write a program to print the ASCII value and the two adjacent characters of the given character.
Input
Ε
Output
69
DF
Answer: (penalty regime: 0 %)
     1 #include<stdio.h>
         int main()
     2
    3 ▼ {
              char a;
scanf("%c", &a);
printf("%d\n", a);
printf("%c ", (a-1));
printf( "%C", (a+1));
printf( "%C", (a+1));
     4
    6
     8
     9
               return 0;
   10 }
```

```
Input Expected Got

Input
```

Finish review