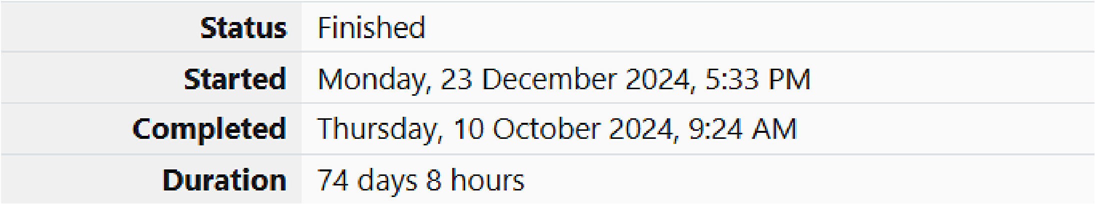
Week 1 – 1:

--Coding-C-Language Features

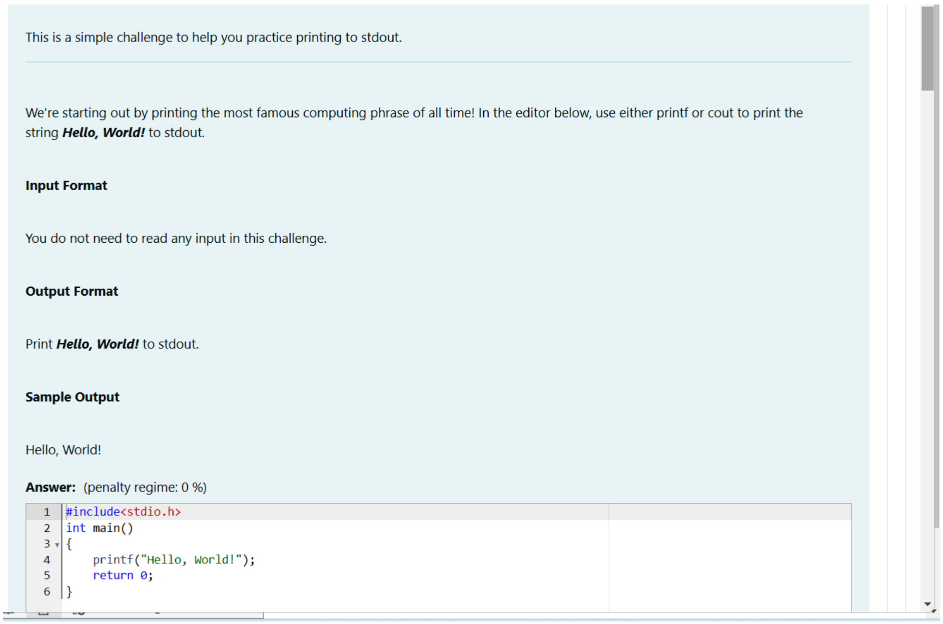
ROLL NO.:240801150

Name: Karthikeyan.M

Q1) 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, World! to stdout. Input Format You do not need to read any input in this challenge. Output Format Print Hello, World! to stdout. Sample Output Hello, World! Answer:(penalty regime: 0 %)



Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 10 October 2024, 9:24 AM
Duration	74 days 8 hours



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, World!*** to stdout.

Input Format

You do not need to read any input in this challenge.

Output Format

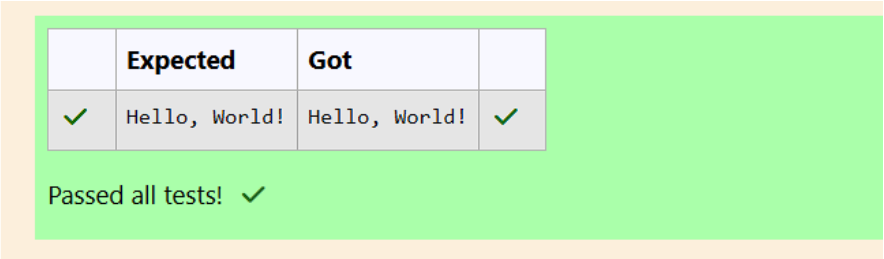
Print ***Hello, World!*** to stdout.

Sample Output

Hello, World!

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    printf("Hello, World!");
    return 0;
}
```



	Expected	Got	
✓	Hello, World!	Hello, World!	✓

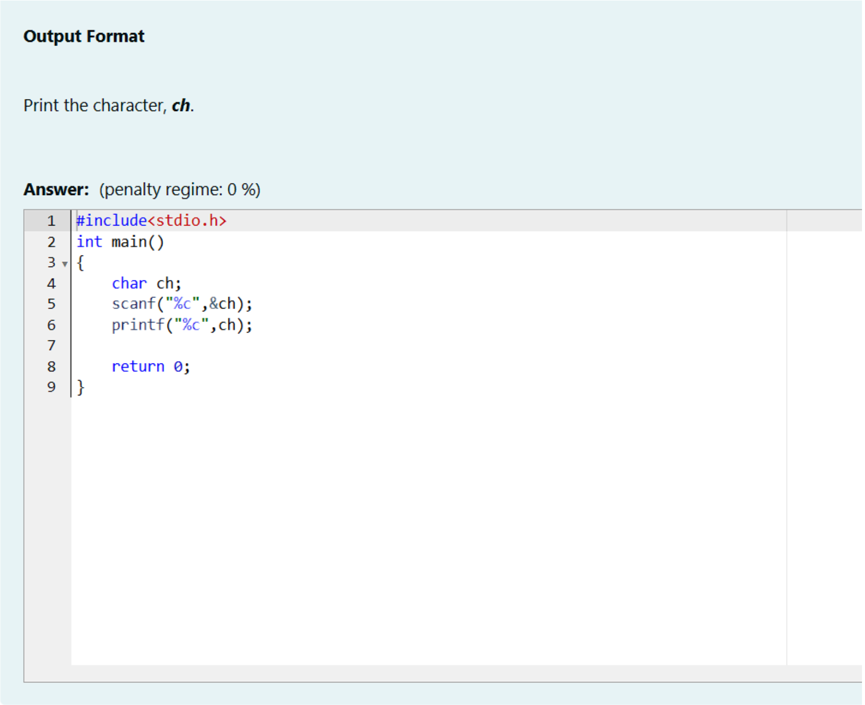
Passed all tests! ✓

Q2)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.

Output Format Print the character, ch.

Answer:(penalty regime: 0 %)

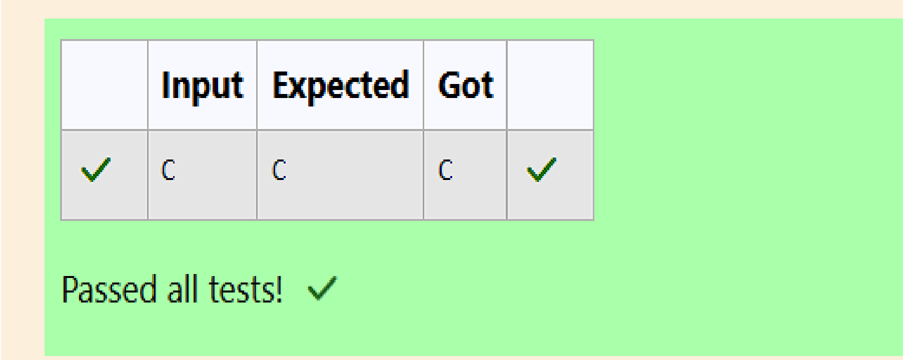


Output Format

Print the character, ***ch***.

Answer: (penalty regime: 0 %)

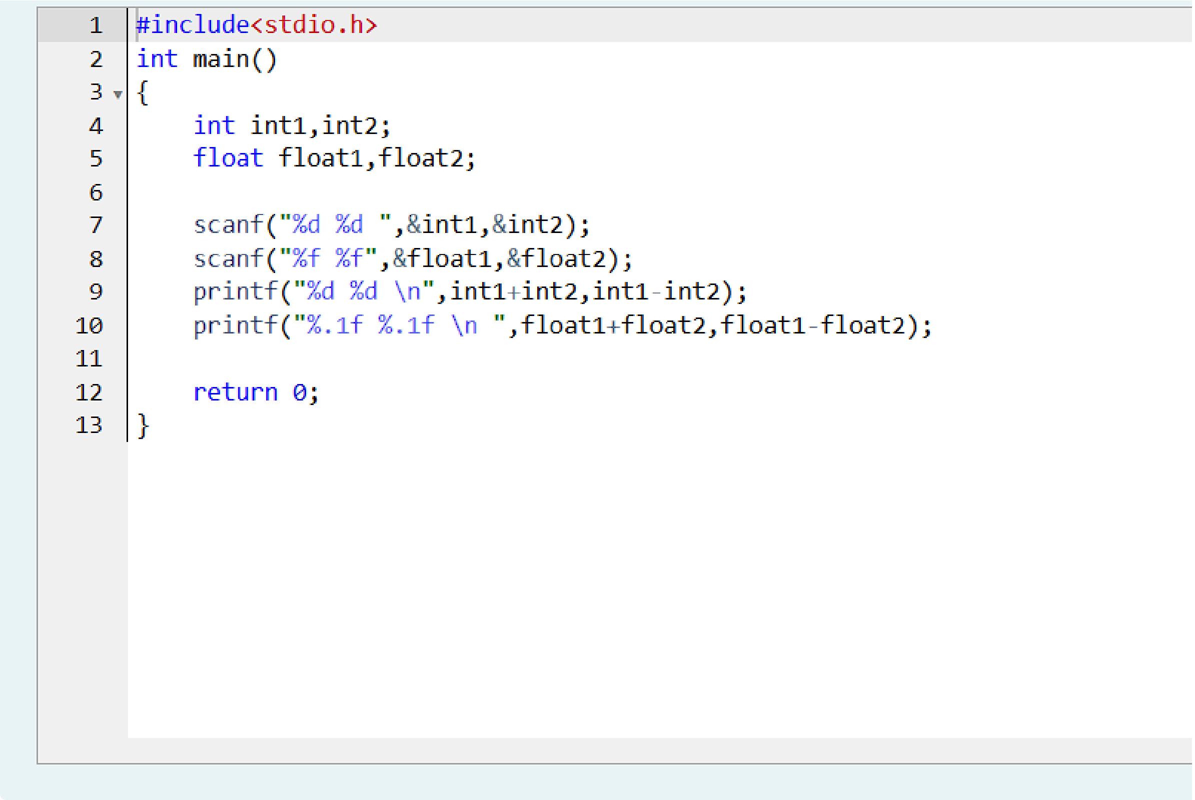
```c
#include<stdio.h>
int main()
{
    char ch;
    scanf("%c",&ch);
    printf("%c",ch);

    return 0;
}
```



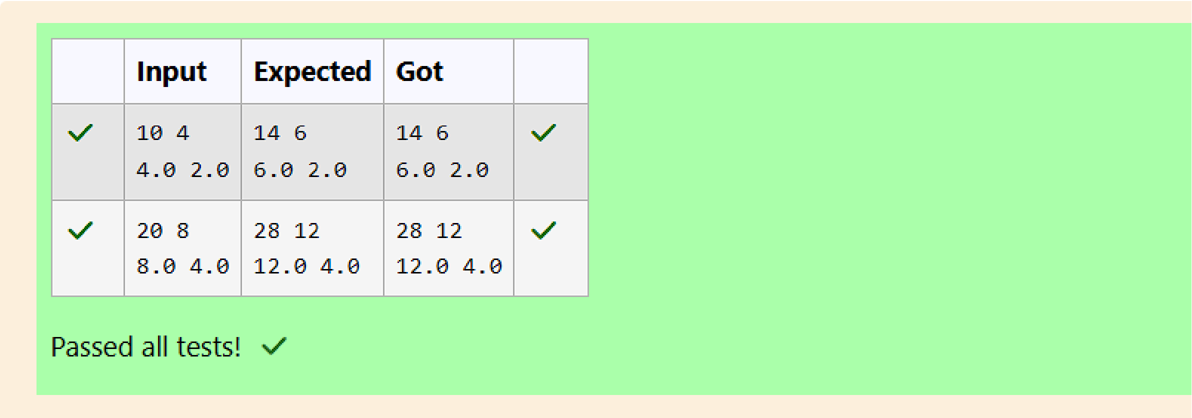
	Input	Expected	Got	
✓	C	C	C	✓

Passed all tests! ✓

Q3) Question text 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 n and m are the two integers



```c
#include<stdio.h>
int main()
{
    int int1,int2;
    float float1,float2;

    scanf("%d %d ",&int1,&int2);
    scanf("%f %f",&float1,&float2);
    printf("%d %d \n",int1+int2,int1-int2);
    printf("%.1f %.1f \n ",float1+float2,float1-float2);

    return 0;
}
```



	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓
✓	20 8 8.0 4.0	28 12 12.0 4.0	28 12 12.0 4.0	✓

Passed all tests! ✓