Name: Sarvesh.R

Reg No: 240701478

Week 1: Overview of C, Constants, Variables and Data Types

1. Say "Hello, World!" With C

Problem Statement:

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 1 Hello, World!

Program:

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Sum and Difference of Two Numbers

Problem Statement:

2.

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:
    - Print the sum and difference of two int variable on a new line.
    - 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 10 4 4.0 2.0

Sample Output 14 6 6.0 2.0

Program:

```c
#include<stdio.h>
int main()
{
    int a,b;
    float c,d;
    scanf("%d%d",&a,&b);
    printf("%d %d",a+b,a-b);
    scanf("%f %f",&c,&d);
    printf("\n%0.1f %0.1f",c+d,c-d);
    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! ✓

Average Marks

Problem Statement:

3.

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format. Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section. All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

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 3 4 6

Sample Output 1 : A 4

Program:

```c
#include<stdio.h>
int main()
{
    int m1,m2,m3;
    char name;
    scanf("%c\n%d %d %d",&name,&m1,&m2,&m3);
    int avg=(m1+m2+m3)/3;
    printf("%c\n%d",name,avg);
    return 0;
}
```

	Input	Expected	Got	
✓	A 3 4 6	A 4	A 4	✓
✓	T 7 3 8	T 6	T 6	✓
✓	R 0 100 99	R 66	R 66	✓

Basic data type

Problem statement:

4.

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);

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. Sample Input

3

12345678912345

a

334.23

14049.30493 Sample Output

3

12345678912345

a

334.230

14049.304930000

Program:

```c
#include<stdio.h>
int main()
{
    int a;
    long b;
    char x;
    float d;
    double e;
    scanf("%d %ld %c %f %lf",&a,&b,&x,&d,&e);
    printf("%d\n%ld\n%c\n%0.3f\n%0.9lf",a,b,x,d,e);
    return 0;
}
```

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

Passed all tests! ✓

5.ASCII Value and Adjacent Characters

Problem Statement:

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input Format:

Reads the character

Output Format:

First line prints the ascii value, second line prints the previous character and next character of the input character Sample Input 1:

E

Sample Output 1:

69

D F

Program:

```c
#include<stdio.h>
int main()
{
    char x;
    int a;
    scanf("%c",&x);
    a=x;
    printf("%d\n",a);
    printf("%c %c",x-1,x+1);
    return 0;

}
```

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests! ✓