Playing With Characters

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.

You can take a string as input in C using scanf("%s", s). But, it accepts string only until it finds the first space.

In order to take a line as input, you can use scanf("%[^\n]%*c", s); where s is defined as char s[MAX_LEN] where MAX_LEN is the maximum size of s . Here, [] is the scan set character. ^\n stands for taking input until a newline isn't encountered. Then, with this %*c, it reads the newline character and here, the used * indicates that this newline character is discarded.

Note: The statement: scanf("%[^\n] %*c", s); will not work because the last statement will read a newline character, \n, from the previous line. This can be handled in a variety of ways. One way is to use scanf("\n"); before the last statement.

Task

You have to print the character, ch , in the first line. Then print s in next line. In the last line print the sentence,sen .

Input Format

First, take a character, ch as input.
Then take the string, s as input.
Lastly, take the sentence sen as input.

Constraints

Strings for and will have fewer than 100 characters, including the newline.

Output Format

Print three lines of output. The first line prints the character, ch.
The second line prints the string,s .
The third line prints the sentence,sen .

Sample Input 0

```
C
Language
Welcome To C!!
```

Sample Output 0

```
C
Language
Welcome To C!!
```

Sum and Difference of Two Numbers

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 .

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.

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 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

- $1 \leq$ integer variables $\leq 10^4$
- $1 \leq$ float variables $\leq 10^4$

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

Functions in C

Objective

In this challenge, you will learn simple usage of functions in C. Functions are a bunch of statements grouped together. A function is provided with zero or more arguments, and it executes the statements on it. Based on the return type, it either returns nothing (void) or something.

A sample syntax for a function is

```
return_type function_name(arg_type_1 arg_1, arg_type_2 arg_2, ...) {
    ...
    ...
    ...
    [if return_type is non void]
        return something of type `return_type`;
}
```

For example, a function to read four variables and return the sum of them can be written as

```
int sum_of_four(int a, int b, int c, int d) {
    int sum = 0;
    sum += a;
    sum += b;
    sum += c;
    sum += d;
    return sum;
}
```

+= : Add and assignment operator. It adds the right operand to the left operand and assigns the result to the left operand.

a += b is equivalent to a = a + b;

Task

Write a function int max_of_four(int a, int b, int c, int d) which reads four arguments and returns the greatest of them.

Note

There is not built in max function in C. Code that will be reused is often put in a separate function, e.g. int max(x, y) that returns the greater of the two values.

Input Format

Input will contain four integers a,b,c,d one on each line.

Output Format

Print the greatest of the four integers.

Note: I/O will be automatically handled.

Sample Input

```
3
4
6
5
```

Sample Output

```
6
```