## HackerRank |

# 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  $s[MAX\_LEN]$  where  $MAX\_LEN$  is the maximum size of s. Here, [] is the scanset character. n stands for taking input until a newline isn't encountered. Then, with this n it reads the newline character and here, the used n 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 s and sen will have fewer than 100 characters, including the newline.

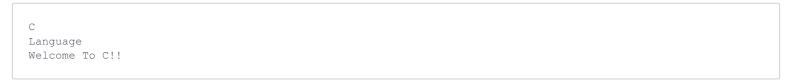
#### **Output Format**

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



## Sample Output 0

