Format specifiers in C

The format specifiers are used in C for input and output purposes. Using this concept the compiler can understand that what type of data is in a variable during taking input using the scanf() function and printing using printf() function. Here is a list of format specifiers.

Format Specifier	Type
%c	Character
%d	Signed integer
%e or %E	Scientific notation of floats
%f	Float values
%g or %G	Similar as %e or %E
%hi	Signed integer (short)
%hu	Unsigned Integer (short)
%i	Unsigned integer
%l or %ld or %li	Long
%lf	Double
%Lf	Long double
%lu	Unsigned int or unsigned long
%lli or %lld	Long long
%llu	Unsigned long long
%o	Octal representation
%p	Pointer
%s	String
%u	Unsigned int
%x or %X	Hexadecimal representation
%n	Prints nothing
%%	Prints % character

These are the basic format specifiers. We can add some other parts with the format specifiers. These are like below −

- A minus symbol (-) sign tells left alignment
- A number after % specifies the minimum field width. If string is less than the width, it will be filled with spaces
- A period (.) is used to separate field width and precision

Example

Live Demo

```
#include <stdio.h>
main() {
   char ch = 'B';
   printf("%c\n", ch); //printing character data
   //print decimal or integer data with d and i
   int x = 45, y = 90;
   printf("%d\n", x);
   printf("%i\n", y);
   float f = 12.67;
   printf("%f\n", f); //print float value
   printf("%e\n", f); //print in scientific notation
   int a = 67;
   printf("%o\n", a); //print in octal format
   printf("%x\n", a); //print in hex format
   char str[] = "Hello World";
   printf("%s\n", str);
   printf("%20s\n", str); //shift to the right 20 characters including the string
   printf("%-20s\n", str); //left align
   printf("%20.5s\n", str); //shift to the right 20 characters including the strin
   printf("%-20.5s\n", str); //left align and print string up to 5 character
}
```

Output

```
B
45
90
12.670000
1.267000e+001
103
43
Hello World
Hello World
Hello World
Hello
Hello
```

We can use these format specifiers for the scanf() function also in the same manner. So we can take the input from scanf() like above how we have printed.