Other Format Specifiers in C

Apart from the common format specifiers for integers and floating-point numbers, there are several other format specifiers that are used for handling characters, strings, memory addresses, and even tracking the number of characters printed.

1. %c: Character Format Specifier

The %c format specifier is used for printing or scanning a single character. It reads or writes a single character from a given variable.

2. %s: String Format Specifier

The %s format specifier is used for printing or scanning strings in C. A string is a sequence of characters terminated by a special null character '\0'. This format specifier reads or writes strings from or to variables.

3. %p: Pointer Format Specifier

The %p format specifier is used for printing memory addresses or pointers. In C, pointers are variables that store memory addresses of other variables. To obtain the memory address of a variable, we use the address-of operator &.

Example of %c, %s and %p

C

```c
#include <stdio.h>
int main()
{
    int a = 'g';
    char b[] = "gfg";
    printf("%c %s %p", a, b, &a);
    return 0;
}
```

Output

```
g gfg 0x7ffec6a2fb5f
```

4. %n: Number of Characters Printed

The %n format specifier is unique because it is not used to print any values. Instead, it is used to store the number of characters printed so far by the printf function into an integer variable.

Example of %n

C

```c
#include <stdio.h>
int main()
{
    int x;
    printf("abc %n cd \n", &x);
    printf("The value of x is %d", x);
    return 0;
}
```

Output

```
abc  cd
The value of x is 4
```

In this example, **%n** stores the number of characters printed up to that point in the variable **x**. first prinf prints "abc " before encountering **%n** so it stores 4 (the number of characters printed so far including whitespace character) in the variable x.