



Tutorial Link <https://codequotient.com/tutorials/Basic Input Output in C - Characters/5a01b3dacbb2fe34b77750b7>

## TUTORIAL

# Basic Input Output in C - Characters

## Chapter

### 1. Basic Input Output in C - Characters

When we write programs, we generally require some input from user and must provide outputs to the user. C language does not provide any keyword for receiving input or producing output. Instead these things will be done using library functions. Following are some example functions for reading/writing various kind of data in our program.

**Reading characters:** The simplest functions to read/write data are `getchar()` and `putchar()`. The `getchar()` function will wait for a character to be typed from keyboard and then return it. The `putchar()` function will print a character at current position on screen. The prototypes for these functions are: -

```
int getchar(void);  
int putchar(int c);
```

They work with characters in their ASCII form, so they use integer variables. If there is any error then `getchar()` will return EOF which is generally equal to -1. Following program will use them: -

```
#include <stdio.h>
```

```
int main()
{
    char ch;
    printf("Enter a character : ");
    ch = getchar();
    printf("The entered character is: ");
    putchar(ch);
    return 0;
}
```

Because the `getchar()` function is implemented in an not obvious manner in C i.e. it will return a single character only after seeing an ENTER from keyboard, so if user type 3 characters and then hit ENTER, in this case `getchar()` leaves the other two characters in buffer. Also the first `getchar()` function will return 2 bytes, 1st byte is the character returned and 2nd byte the ENTER key. This creates annoying situations when there is a sequence of `getchar()` functions. So some other functions are defined like `getch()` or `getche()` in "conio.h" header to facilitate the programmer. So if you run following program which must ask three characters 1 by 1 from user, may not work perfectly if the user enters more characters in first go: -

```
#include <stdio.h>
int main()
{
    char ch1, ch2, ch3;
    printf("Enter a character : ");
    ch1 = getchar();
    printf("\nThe entered character is: ");
    putchar(ch1);
    // Second character
    printf("\nEnter a character : ");
    ch2 = getchar();
    printf("\nThe entered character is: ");
    putchar(ch2);
    // Third character
    printf("\nEnter a character : ");
    ch3 = getchar();
    printf("\nThe entered character is: ");
    putchar(ch3);
    return 0;
}
```

In this program if user enters a multiple characters in first line, then next `getchar()` function will not wait for keyboard, instead they read from buffer and it will not be a friendly situation in programs. Try running this program with different inputs and learn by yourself.

