

C Input and Output

Input means to provide the program with some data to be used in the program and **Output** means to display data on screen or write the data to a printer or a file.

`scanf()` and `printf()` functions

The standard input-output header file, named `stdio.h` contains the definition of the functions `printf()` and `scanf()`, which are used to display output on screen and to take input from user respectively.

```
void main()
{
    // defining a variable

    int i;

    printf("Please enter a value...");

    scanf("%d", &i);

    printf( "\nYou entered: %d", i);
}
```

Format String	Meaning
%d	Scan or print an integer as signed decimal number i=10
%f	Scan or print a floating point number i=10.5
%c	To scan or print a character A,B,C
%s	To scan or print a character string. The scanning ends at whitespace. ' ABC '

getchar() & putchar() functions

The **getchar()** function reads a character from the terminal and returns it as an integer. This function reads only single character at a time.

The **putchar()** function displays the character passed to it on the screen and returns the same character. This function too displays only a single character at a time.

This function too displays only a single character at a time. In case you want to display more than one characters, use **putchar()** method in a loop.

```
#include <stdio.h>

void main( )
{
    int c;

    printf("Enter a character");

    c = getchar(); //S

    putchar(c); //S
}
```

`gets()` & `puts()` functions

The `gets()` function reads a line from **stdin**(standard input)

The `puts()` function writes the string `str` and a trailing newline to **stdout**.

```
#include<stdio.h>

void main()
{

    char str[100]; char a[10];

    printf("Enter a string");

    gets( str ); // I am a Student
```

```
puts( str ); //I am a Student  
  
getch();  
  
}
```

Difference between scanf() and gets()

The main difference between these two functions is that `scanf()` stops reading characters when it encounters a space, but `gets()` reads space as character too.

If you enter name as **Study Tonight** using `scanf()` it will only read and store **Study** and will leave the part after space. But `gets()` function will read it completely.

a= 10, I am a student