PROBLEM SOLVING AND C PROGRAMMING

I YEAR – I SEM

UNIT – II C PROGRAMMING BASICS

TOPIC – MANAGING INPUT & OUTPUT OPERATIONS

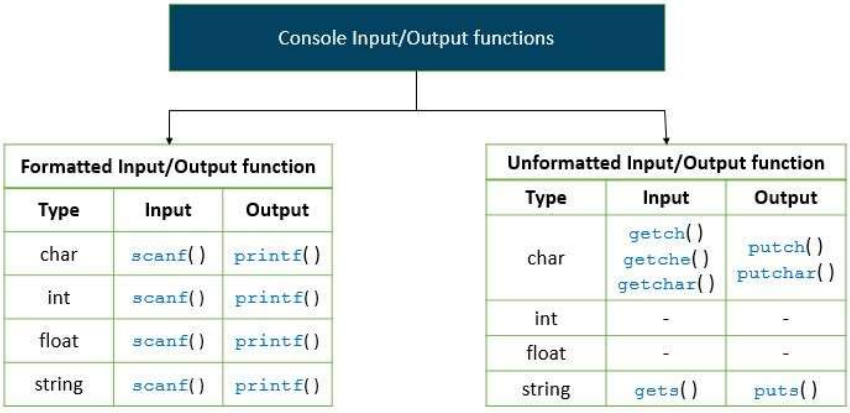
If the primary advantage of computers is their ability to communicate with the user during program execution (i.e.) computer communicator with the users by means of input/output devices i/o devices accomplish i/o operations. Input operation involves data transfer from the input device to the computer memory and in output operation data transfer is from the computer memory to the output device. This feature enables the programmer to enter the value in to the variables, when the program is entered without altering the program.

In C language two types of Input / Output statements are available. Several functions are available for Input/Output operations in 'C'. These functions are collectively known as the standard I/O Library.

The Input and Output Functions are Divided into Two types.

1) Unformatted Input / Output Statements

2) Formatted Input / Output Statements



Formatted Input/Output function

Type	Input	Output
char	scanf( )	printf( )
int	scanf( )	printf( )
float	scanf( )	printf( )
string	scanf( )	printf( )

Unformatted Input/Output function

Type	Input	Output
char	getch( ) getche( ) getchar( )	putch( ) putchar( )
int	-	-
float	-	-
string	gets( )	puts( )

Unformatted Statement

This input & output statements works only with character data type. Do not required any conversion symbol for identification of data type because it works out with character data type.

getc()

This is used to accept a single character from the standard input to a character variable.

The general form of getc() is

Character_variable = getc()

putc()

This is used to display a single character in a character variable to standard output device.

The general format for using this function is

putc(character variable)

getchar()

The simplest of all input operations is reading of a single character from the standard input unit. Reading a single character can be carried out by a function called getchar. This function does not require any arguments, though a pair of empty parenthesis follows the word getchar().

The general from of getchar() as

Variable_name = getchar() ;

putchar()

The function putchar() writes its character argument on the screen each time it is called.

The general format for using this function is:

putchar(variable);

Example Program :

```
#include<stdio.h>
#include<conio.h>
void    main()
{
char c ;
c =      getchar() ;
putchar(c) ;          /* displays the character entered */
putchar(66)         ; /* displays the character B */
```

putchar(‘a’) ; /* displays a */
}

gets()

The gets function is used to receive a string from the standard inputdevice. This function accepts a single argument. The argument passed within a function must be a data item which represents a string or a character array. The string received by the gets function may include white space characters. The gets function will terminate with a new line character (press enter key at the end of the string).

The general form is:

gets(argument) ;

puts ()

This function is used to output a string at a time in the standard output device.

The general from is

puts(arguments) ;

Puts function accepts only a single argument

Example Program :

void main()

{

char name[25];

puts(“Enter the name:”);

gets(name);

puts(“\n print the name:”);

puts(name);

}

Output :

Enter the name : xxx

Print the name : xxx

Formatted Statement

This can read all types of data values. They require conversion symbol to identify the data types.

a) The scanf() function

The function scanf() is used to read input values,

the general format is

scanf("control string", var1, var2, … var n) ;

Where
Control string - contains the required formatting specifications enclosed within double quotes,
var1, var2 …. var n -are arguments that represent the individual input data items.

The control string includes conversion specifications directing the conversion of the next input data.

b) The printf() function

Output data can be written form the computer on to a standard output device using the printf function. This function can be used to output any combination of digits, characters and strings. This printf function move the data from the computers memory to the standard output device

The general form is

printf("control string", var1, var2, ….var n) ;

Where
Control string - contains the required formatting specifications enclosed within double quotes,
var1, var2 …. var n -are arguments that represent the individual input data items.