**char** a single byte, capable of holding one character in the local character set.

**int** an integer, typically reflecting the natural size of integers on the host machine.

**float** single-precision floating point.

**double** double-precision floating point.

**short** **int** sh;

**long** **int** counter;

size of data types :

int basically depends on machine type

short is 16 bits in size

long is 32 bits

int is either 16 or 32 bits

Signed and Unsigned is basically for char type

2n wheren is the number of bits in the type

Char are 8 bits

Signed chars are -128 to 127

The type long double : float , double and long double

So if we want to check symbolic constants to all the sizes then we can check these 2 header files

**limits.h**

**float.h**

The complete set of escape sequences is

\a       alert (bell) character  
\b       backspace  
\f       formfeed  
\n       newline  
\r       carriage return  
\t       horizontal tab  
\v       vertical tab  
\\       backslash  
\?       question mark  
\′       single quote  
\"       double quote  
\*ooo*  octal number  
\x*hh*  hexadecimal number

‘\0’ is a null character

Research on things not known:

\f

It skips to the start of the next page. (Applies mostly to terminals where the output device is a printer rather than a VDU.)

>> Article read : <https://stackoverflow.com/questions/4334370/escape-sequence-f-form-feed-what-exactly-is-it>

\r moves to the initial portion of current line

For example :

Helloworld\r

Would give o/p : |Helloworld

\b : backspace

Wrote a sample program to check the output of escape sequence.

#include<stdio.h>

int main()

{

printf("Checking\b how b works\n");

printf("Checking\r how r works\n");

printf("Checking\v how v works\n");

return 0;

}

Checking how b works : Analysis \b removed 1 character

how r works \r skipped one word

Checking

how v works \v shifted words after it to next line

Analyssis

Clearing doubts with \r

baking checking\r how r works\n

o/p

how r worksing

wtvr after \r comes to the first position and then starts printing

**Deep dive on increment operator**

So what’s the difference between ++i and i++

Image I = 1 and j = 1

I = 1;

J = ++I ;

o/p will be I = 1 and j = 2

Image I = 1 and j = 1

I = 1;

J = I++ ;

o/p will be I = 1 and j = 1

Article : <https://stackoverflow.com/questions/24853/c-what-is-the-difference-between-i-and-i>

Sample code for strlen :