# *C language part 2*

1. What is scanf() in c language.

=> scanf() is not a keyword.

=> scanf() is the predefined function.

=> scanf() doing both work print and value user input.

We make first program by using scanf().

$ #include<stdio.h>

$ int main ()

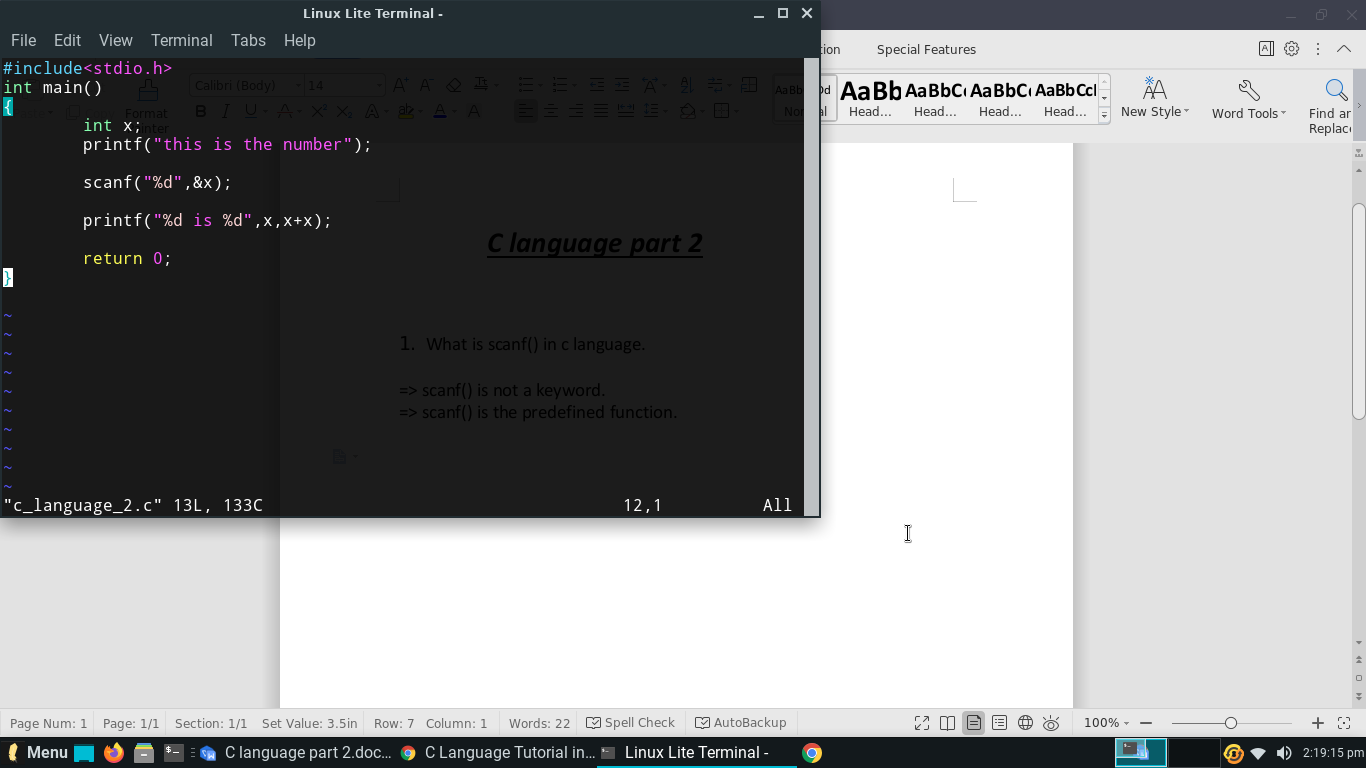
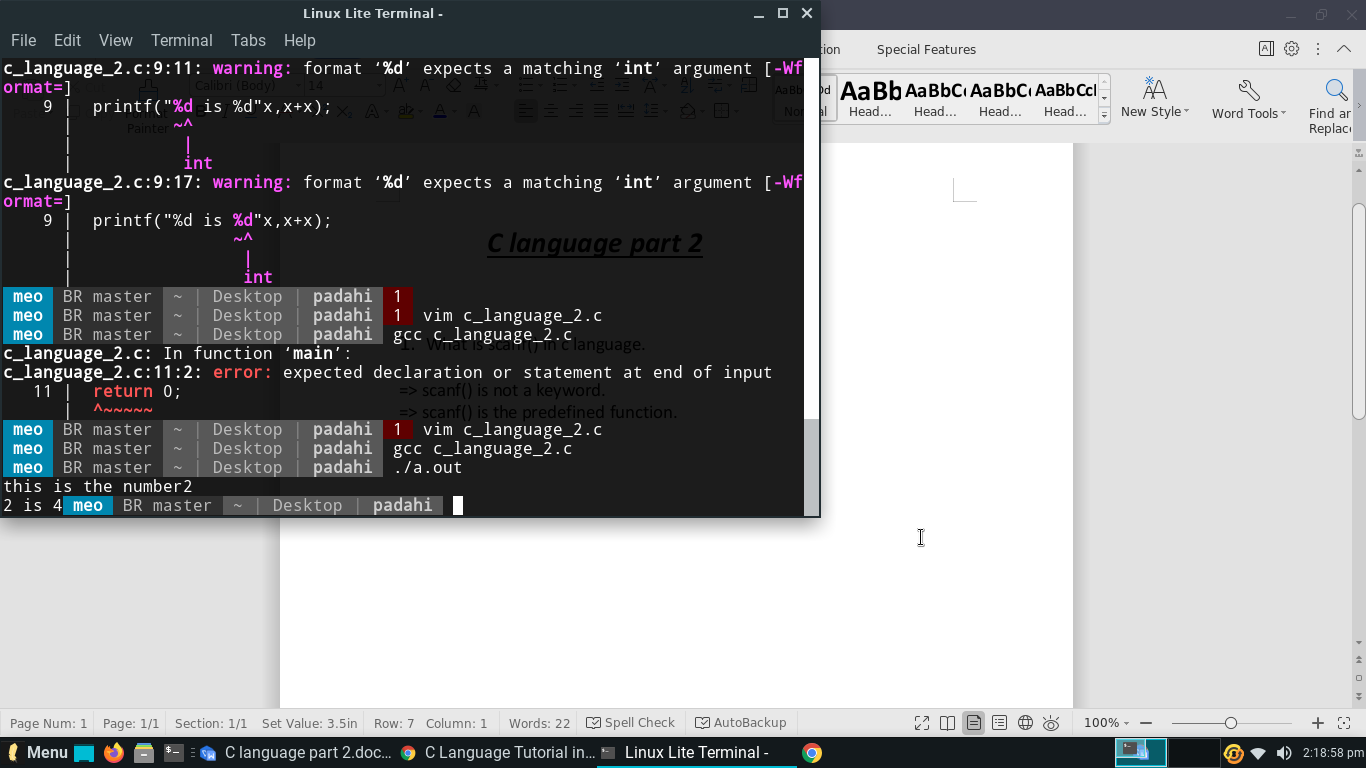
$ {   
$ int x;  
$ printf(“sentences”); (operational)

$ scanf(“%d”,&x);

$ printf(“using the sentence formula”,x,x+x);

$ return 0;

$}

1. what is the ASCII ?

=> Full form of ASCII is “**American Standard Code for Information Interchange**”.

=> ASCAII code is the numerical representation of a character such as ‘a’ or ‘@’ or an action of some sort.

=> ASCII is a technic that the alphabetic is change into number but the number value is fixed.

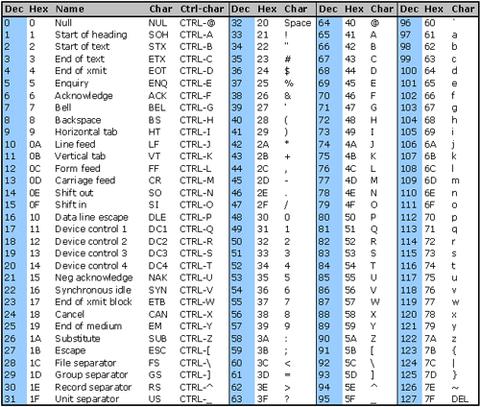
=> The total ASCII code is 256. 128 + 128 .

‘A’ = 65 ‘o’ = 48

‘a’ = 97 ‘1’ = 49

‘b’ = 98 ‘9’ = 57

‘@’ = 64 ‘ ’ = 92

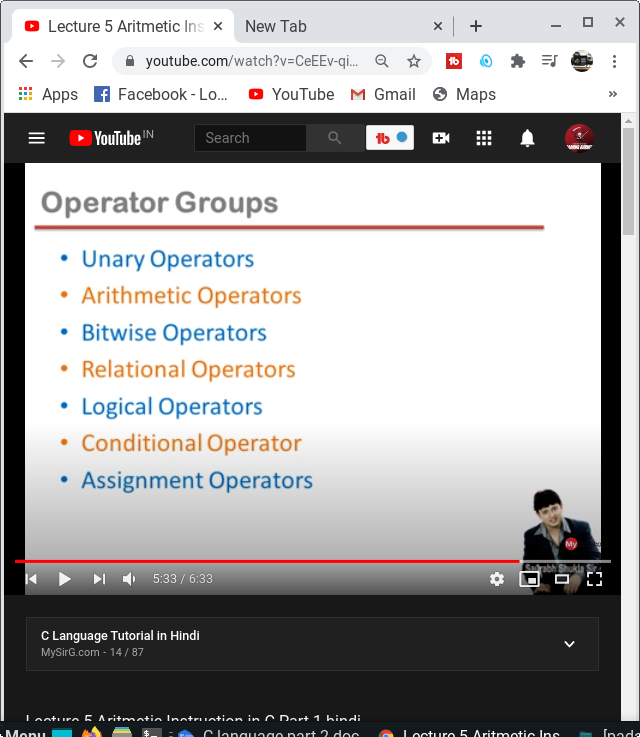


1. What is Arithmetic Instruction in c.

=> An Instruction which is used to manipulate data using operators, is known as Arithmetic Instruction. (manipulate main Hera fare Karina)

=> There is no BODMAS in c language.

=> There is the similarly formula like a BODMAS in the Arithmetic Instruction :-



=> Unary operators :-

+,-,++,--,sizeof()

++ increment operator

$#include<stdio.h>

$int main()

$ {

$ int x=3;

$ x++;

$ printf("%d",x);

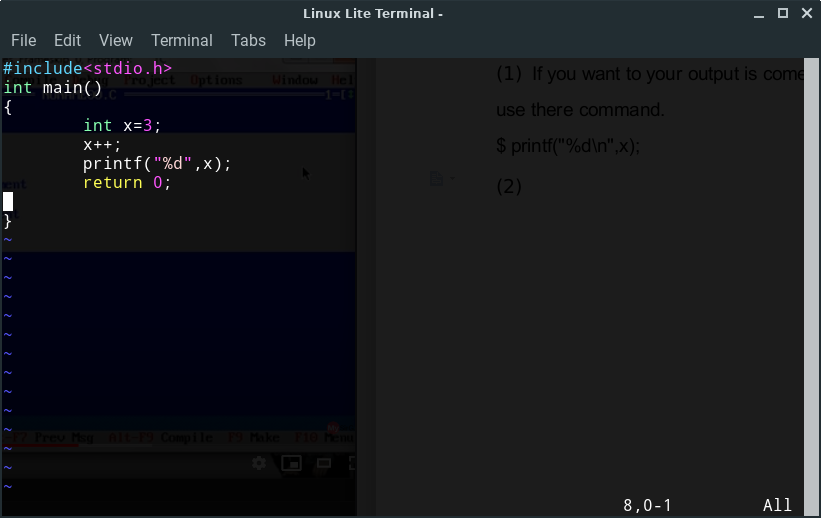
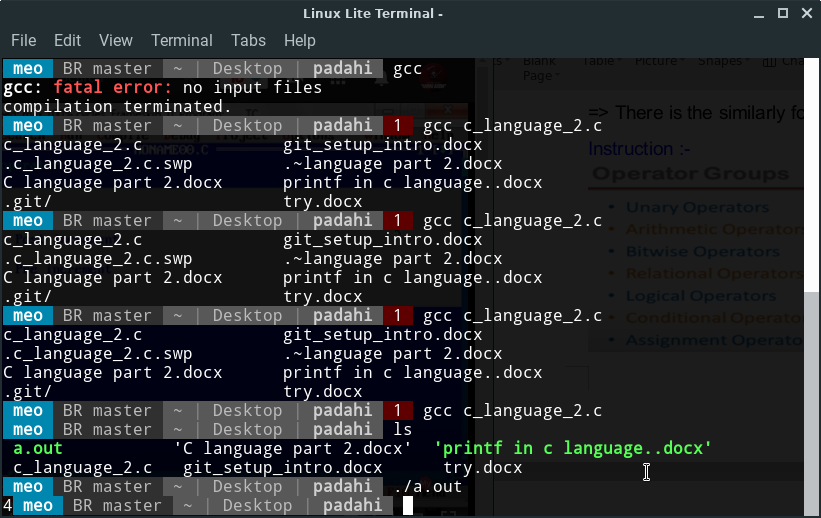
$ ++x;

$ printf("%d",x);

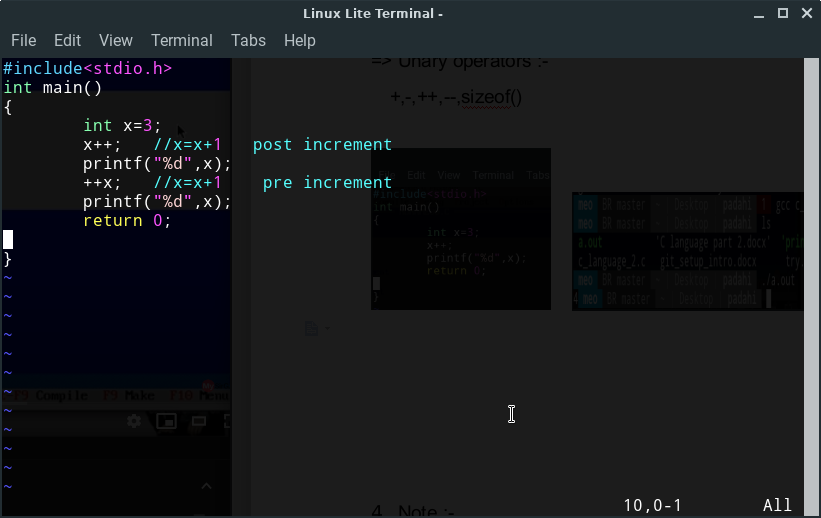
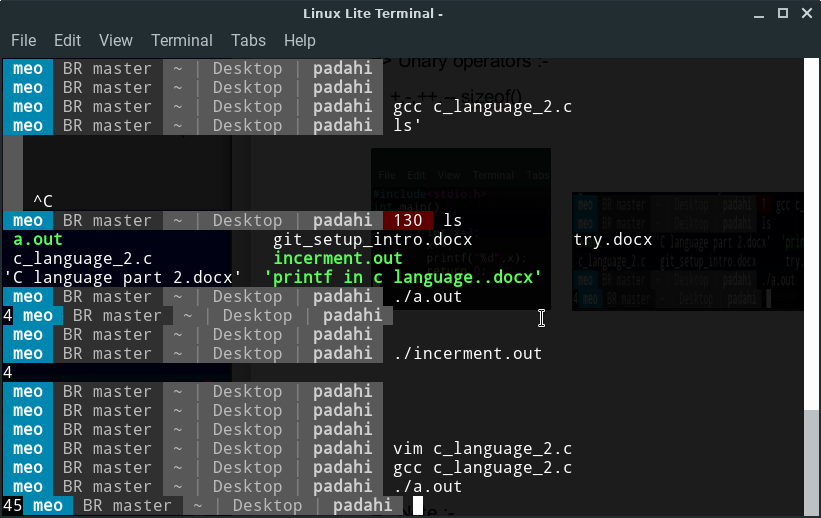
$ return 0;

$ }

1 2

3 4

-- decrement operator

$ #include<stdio.h>

$ int main()

$ {

$ int x=3;

$ x--;

$ printf("%d",x);

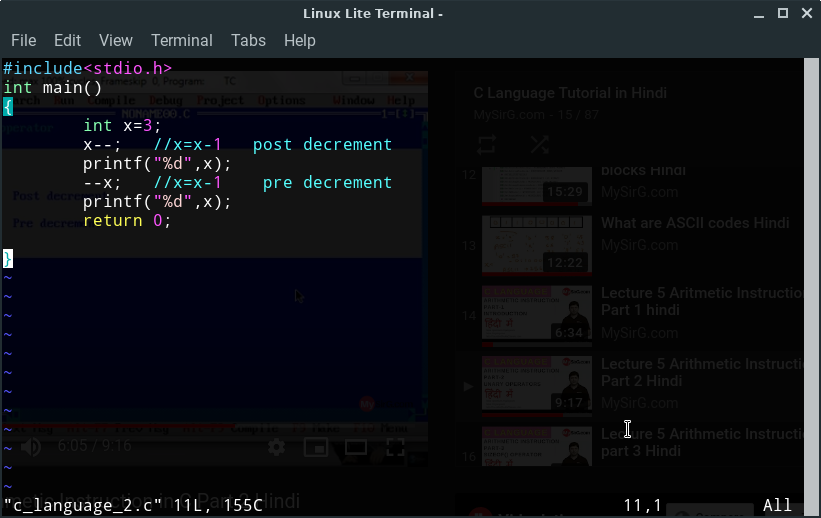
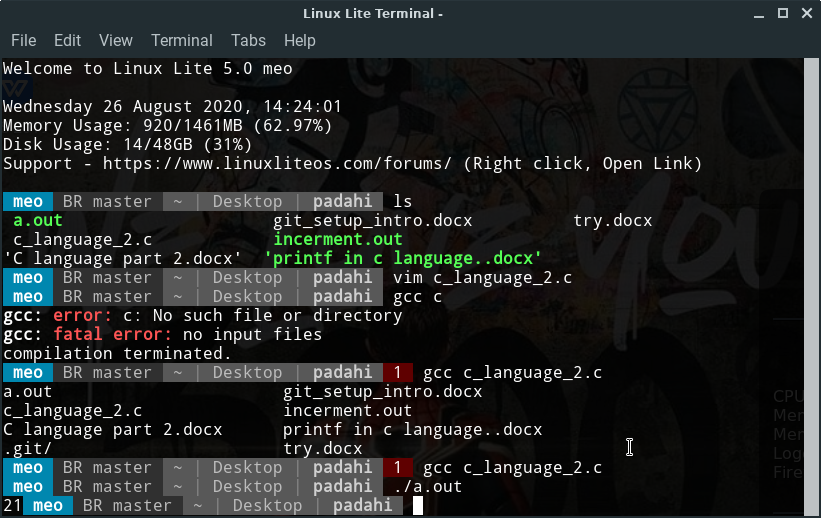
$ --x;

$ printf("%d",x);

$ return 0;

$ }

1 2

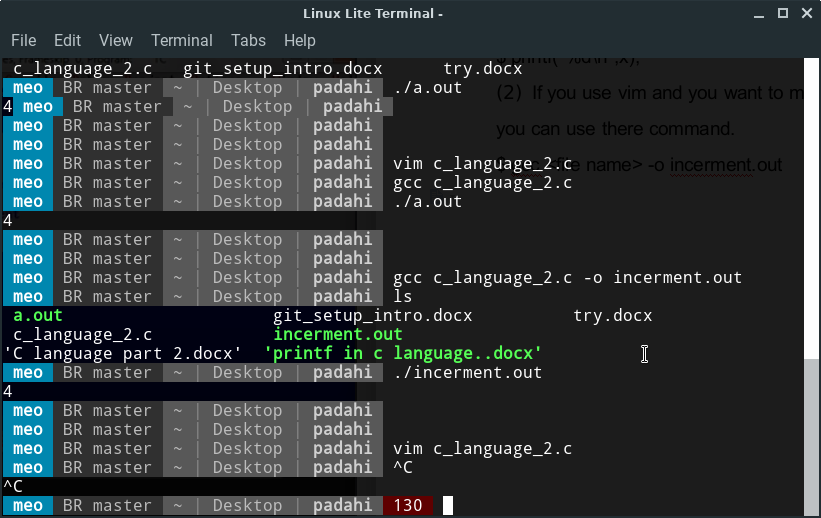
1. Note :-
2. If you want to your output is come under the next line, so you can use there command.

$ printf("%d\n",x);

When you see that your output is come under next line.

1. If you use vim and you want to my output is name is change, so you can use there command.

$ gcc <file name> -o incerment.out



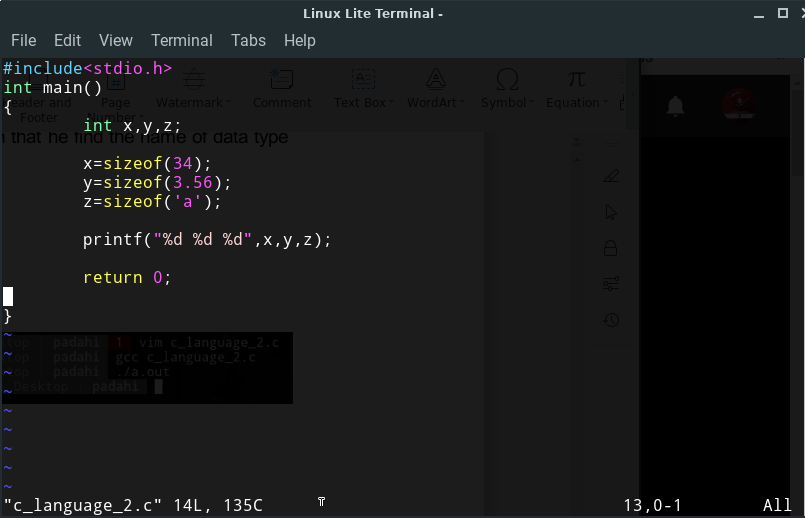
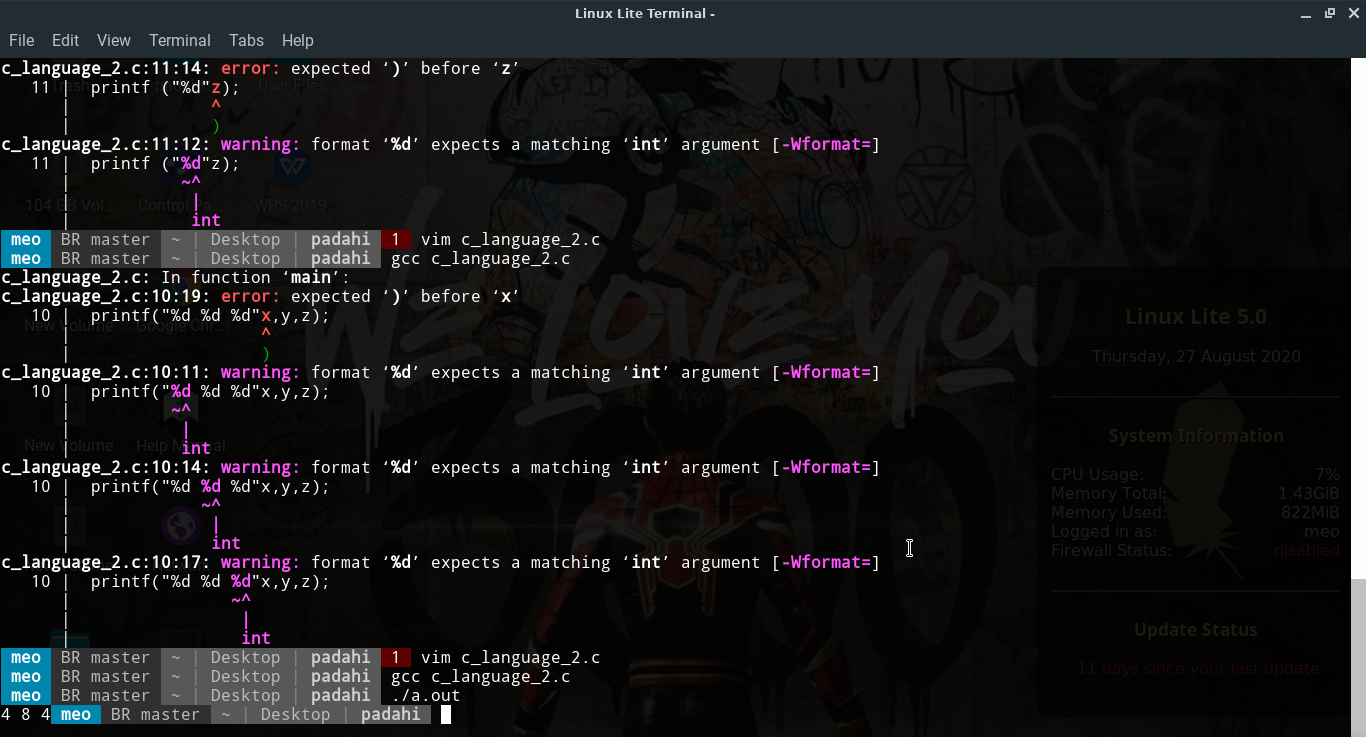
1. What is sizeof in c ?

=> sizeof is the function that he find the name of data type

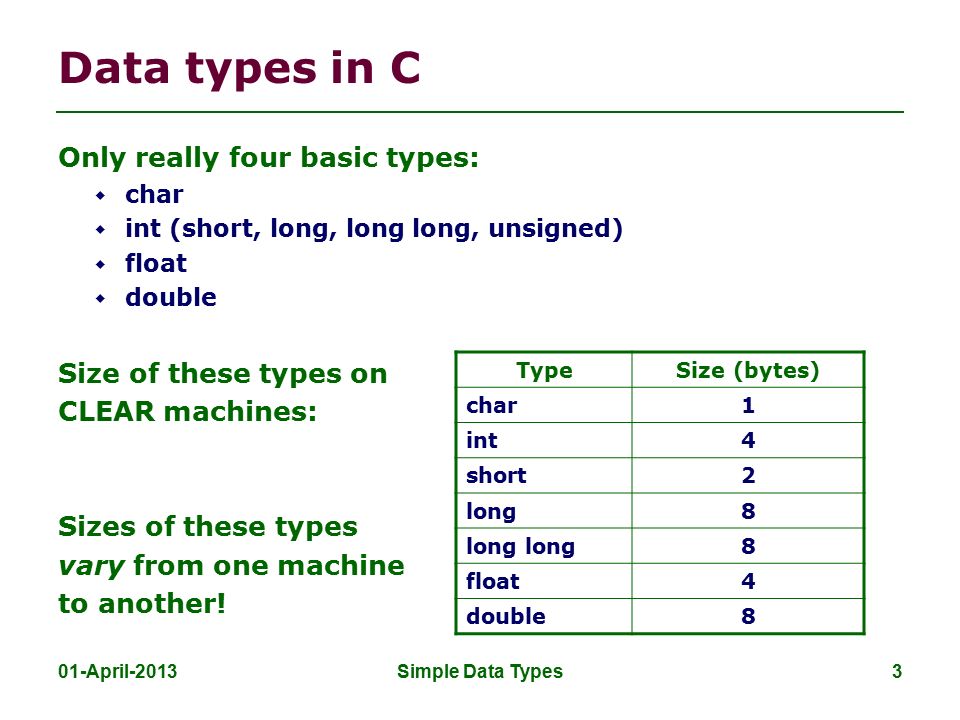
4 => float => 34

8 => double => 3.56

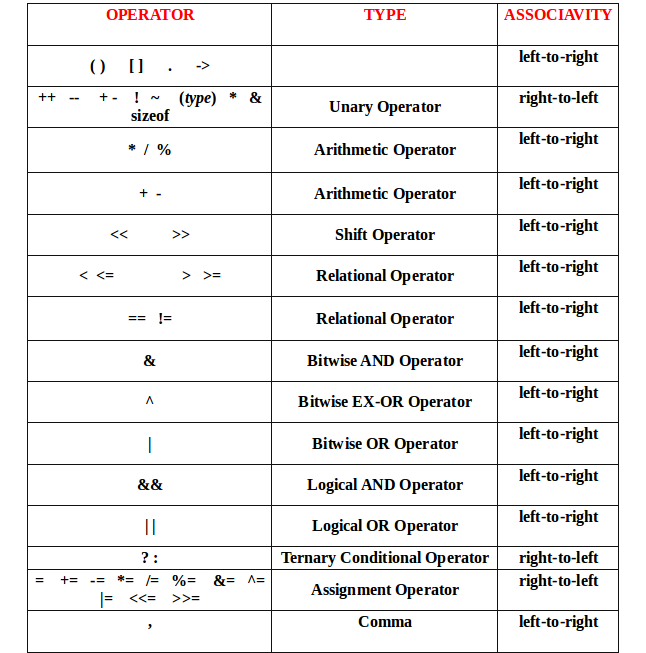
4 => int => a

Note :-

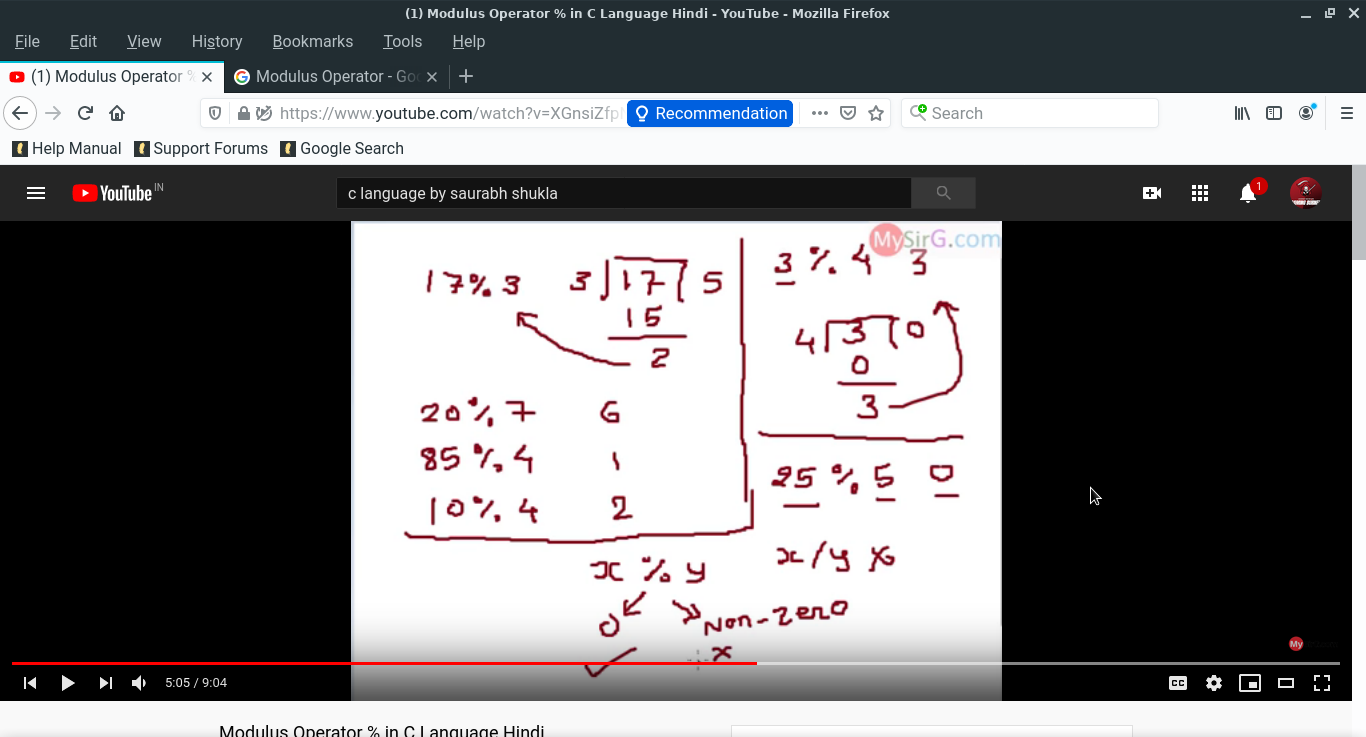
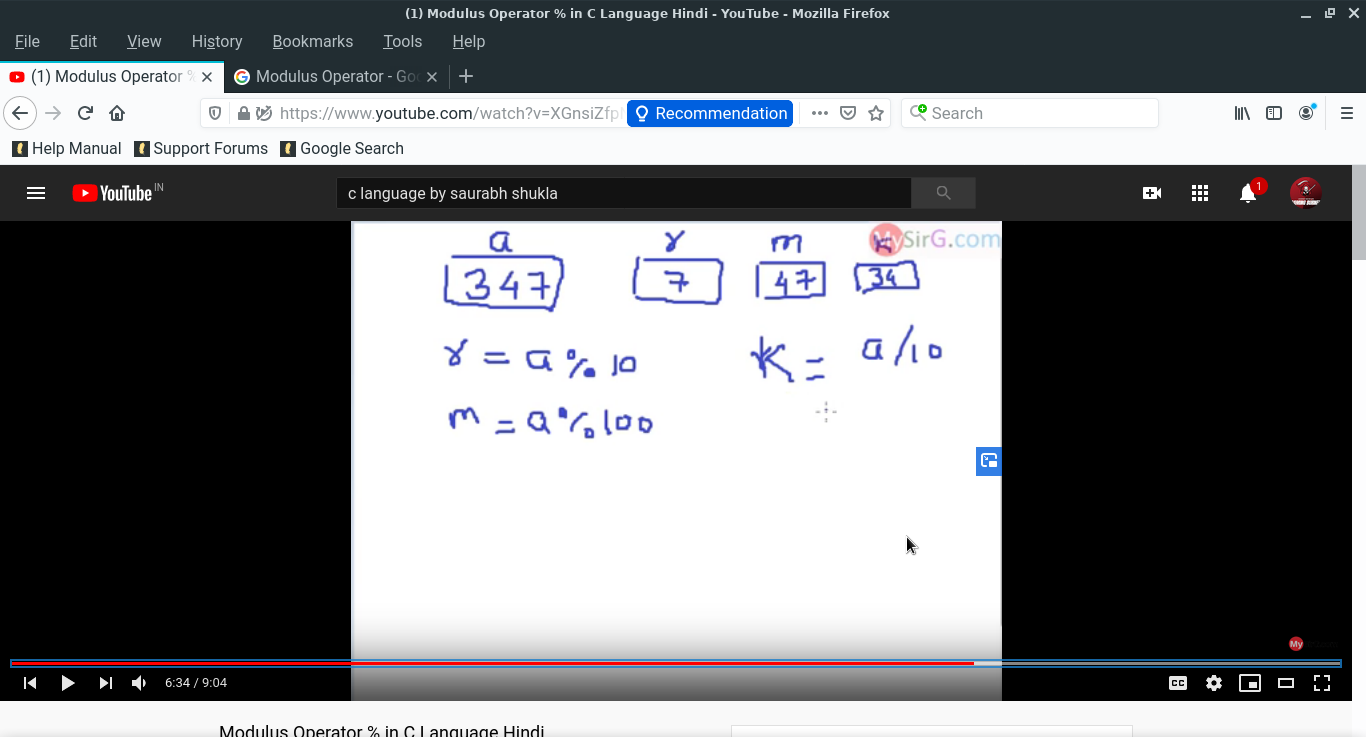


1. There are many type of Operator :-

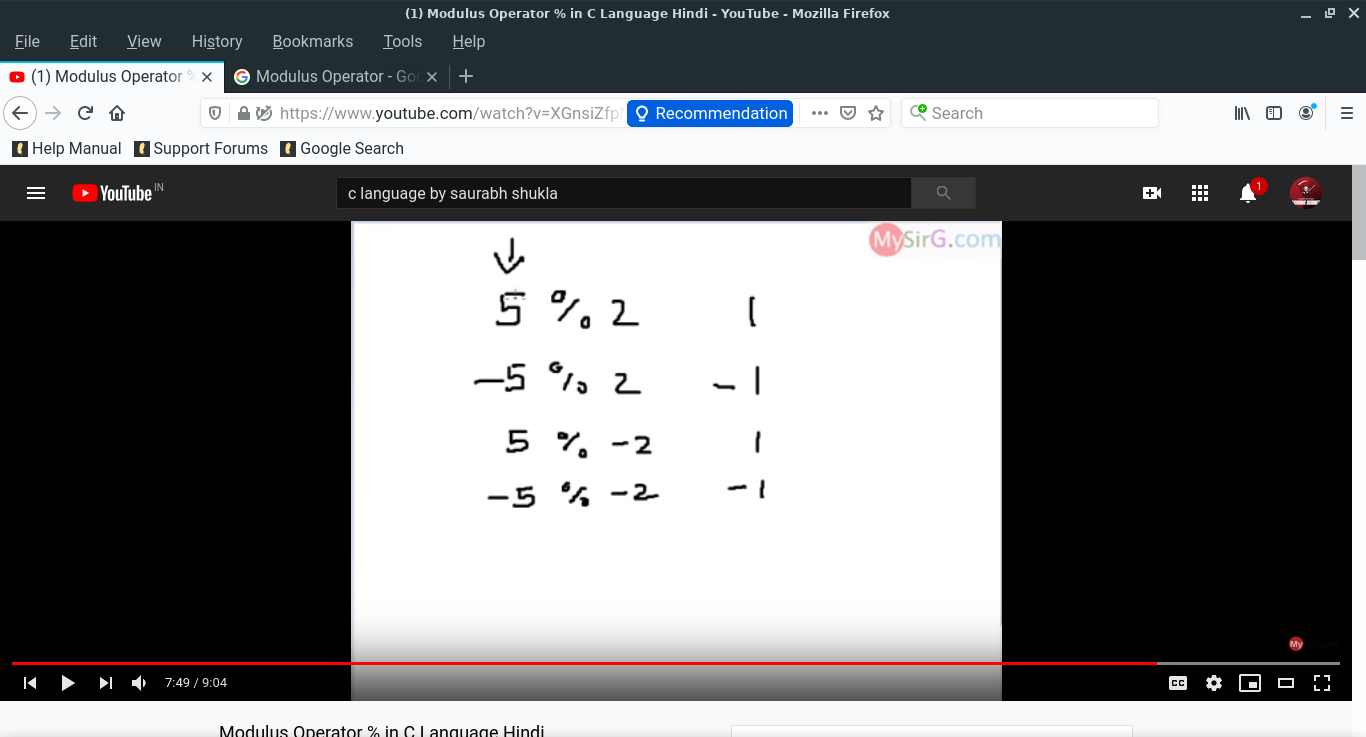
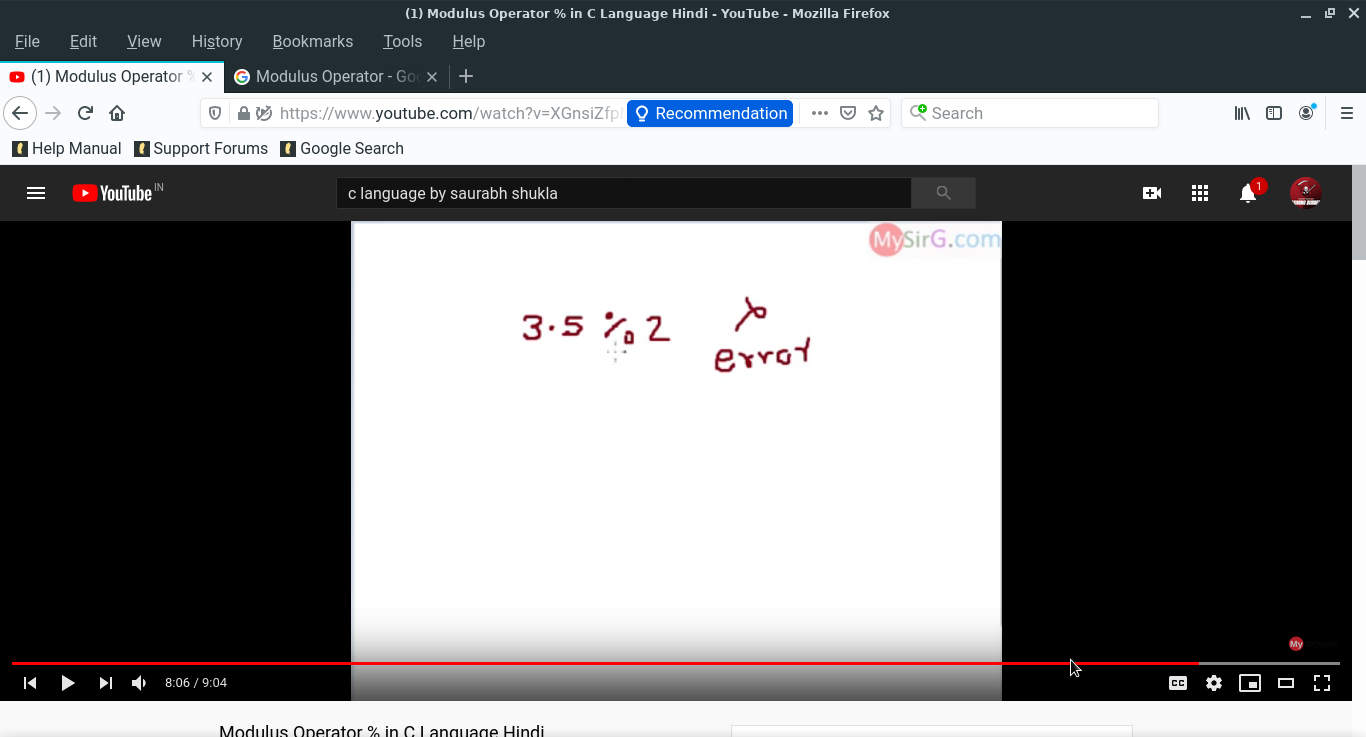


1. What is modulus operator.

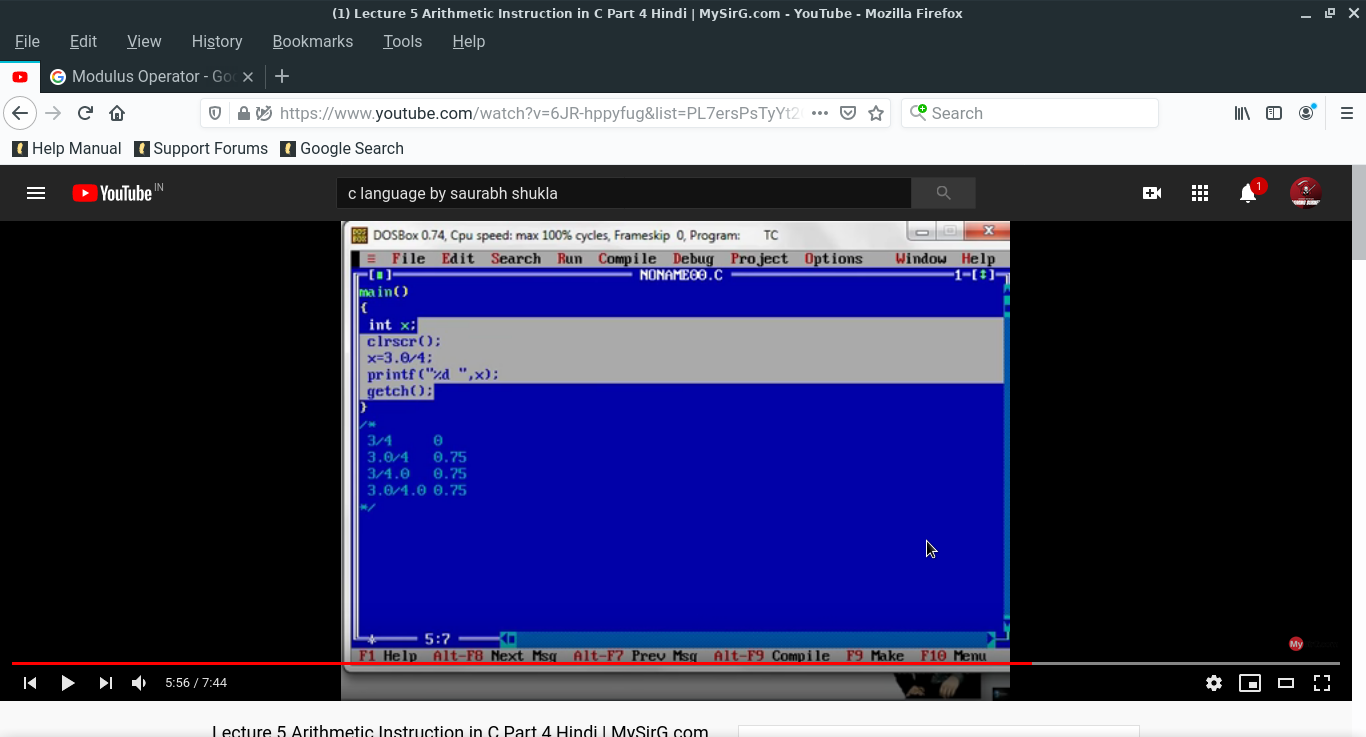
=> Modulus operator give remainder. The modulo operator, denoted by %, is an arithmetic operator. The modulo division operator produces the remainder of an integer division.

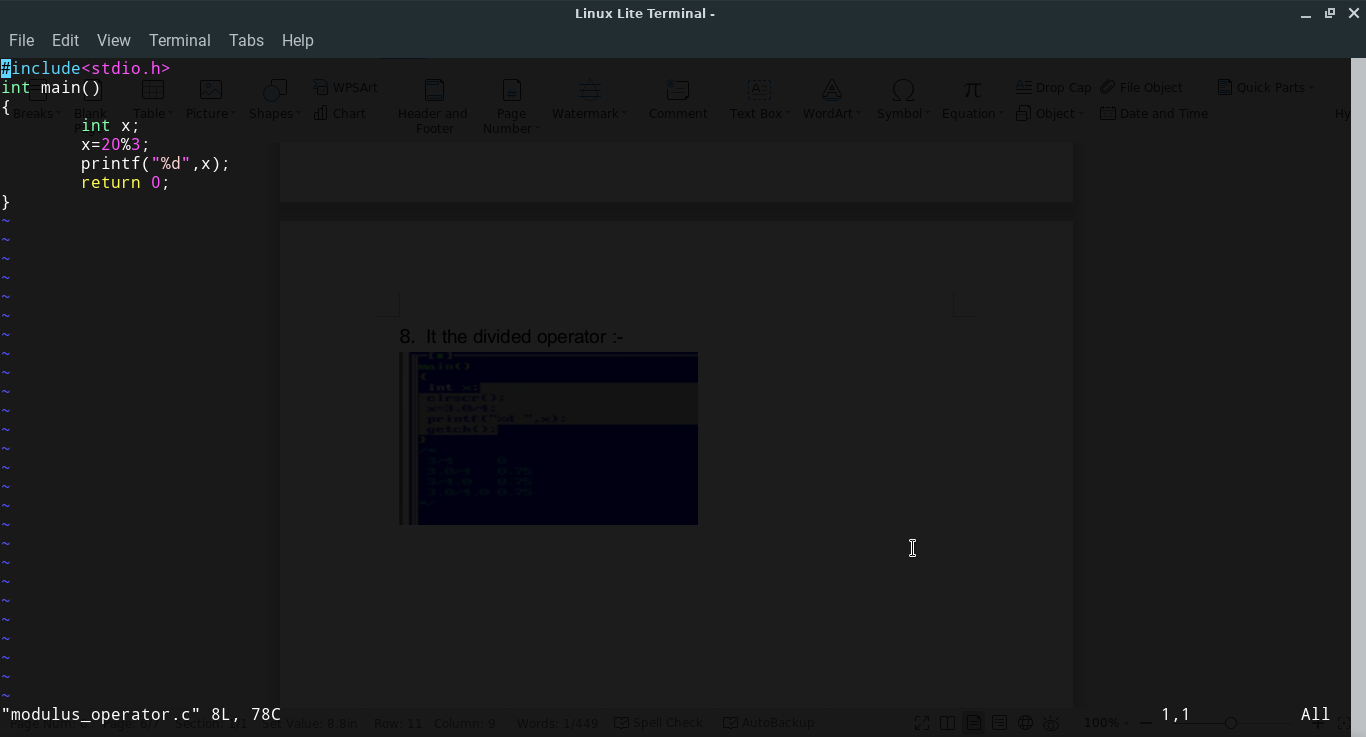
Modulus operator only show left side side and do not show real contents.

=> It the divided operator :-



=> It is the Modulus operator :-



in name of (1)Modulus operator in c program folder.

1. What is Binary language (0,1) in detail.

=> Let you take 2 GB. 0byte to 2 GB

0byte ->1byte ->2byte ->3byte ->-------- ->1 MB ->--------- ->2GB

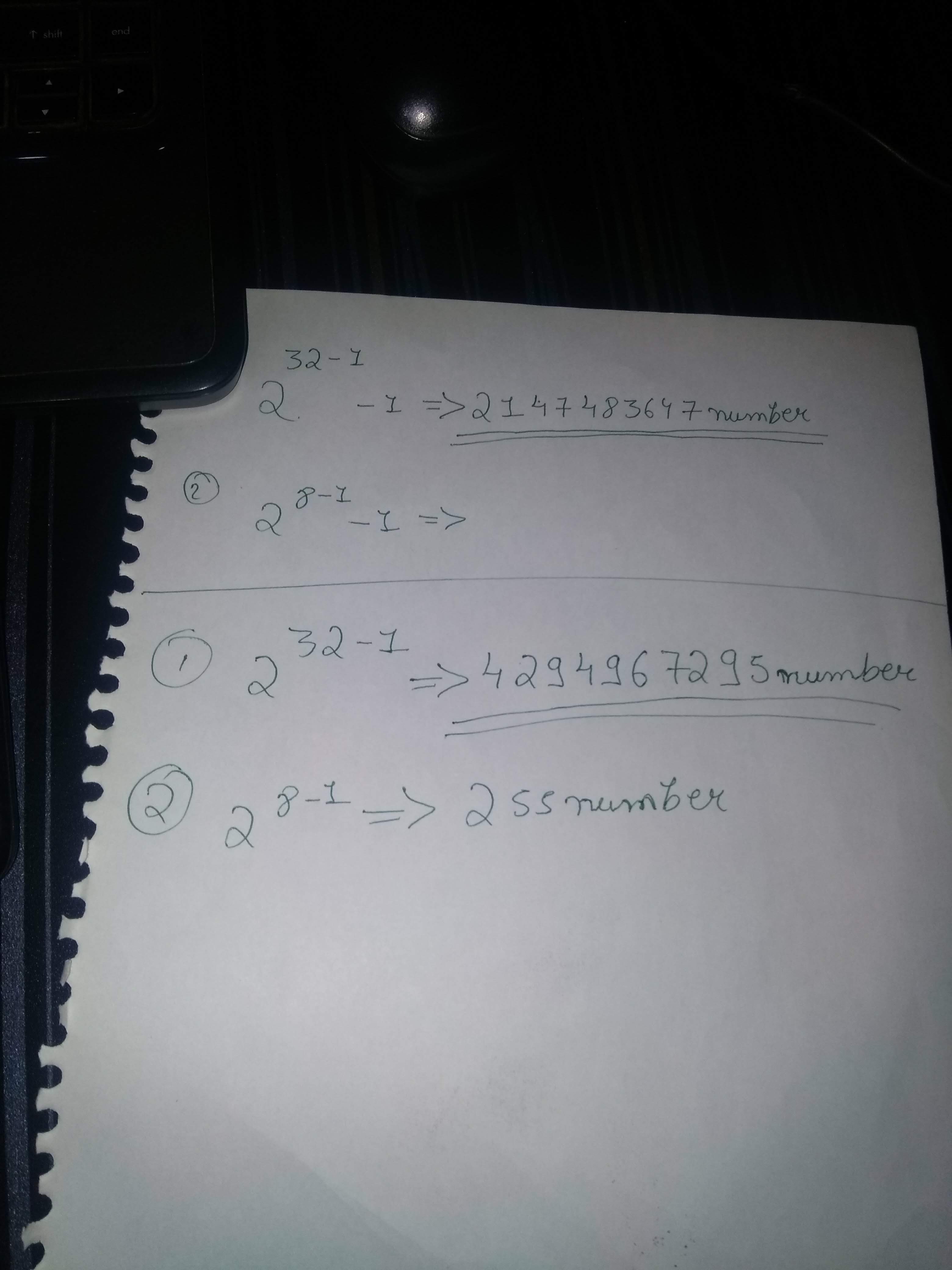
1 byte -> 8bit and it also 8 bit architecture

Suppose we make 46 byte.

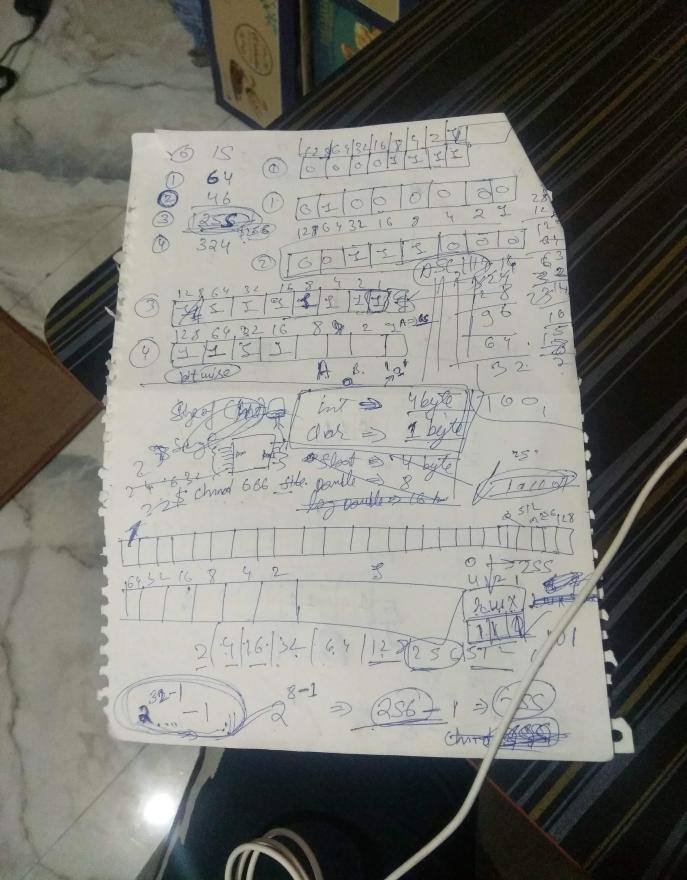
128 64 32 16 8 4 2 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

And if you have <any number> byte and you want to make architecture but you do not know how many mix number represent in block, so use these formula to find the block.



(1) (2)

1. What is Bit wise operator ?

=> Type of Bit wise operator :-

1. AND & “approx say multiple”
2. Or | “approx say sum”
3. NOT ! “invert”
4. XOR(imp) ^

Suppose a = 25 b = 30 c = ans

AND => AND is use to multiple the number, this like

128 64 32 16 8 4 2 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

b =

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

a =

&

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

c =

OR => OR is use to add the number, this like:-

128 64 32 16 8 4 2 1

1 1 <carry>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

b =

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

a =

+

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

c =

Note :- 1+1 =10

1+1+1 =11

NOT => NOT is use to invert the number, this like and NOT is required only one number.

128 64 32 16 8 4 2 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

a =>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

!

!a =>

XOR => XOR is use to

128 64 32 16 8 4 2 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

a =>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

b =>

^

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

c =>

\* All there work and diagram also given in the photo:-

