# **Bits Manipulations**

* **<< is left shift operator. Bits are shifted to left.**
* **>> right shift operator.**
* **~ this is negation operator.**
* **& and | or**
* **1’s and 2’s complement**

1’scomplemnt of 0010 is 1101

2’s complement of 0010 is 1101

+ 1

\_\_\_\_\_

1110

* **Negative nos in pc are stored in form of 2’s complement. If no is of 32 bits then 31st bit is sign bit 0:- positive 1:-negative.**
* Procedure for the negation:-1)first 1’s complement

2)check sign bit

3) if sign bit is 1 do 2’s complement else stop

* **Xor operator(^)**

Even no of 1’s:-0

Odd no of 1’s:-1

* **Swapping of two nos using XOR operator**

int a=5;

int b=6;

**a=a^b;**

**b=a^b;**

**a=a^b;**

* **Check whether bit is set or not**

bool b=(a>>i)&1;//check whether set or not

i= ith bit

* **Set ith bit**

a=a|(1<<i);//set ith bit

for eg 101 then 1<<1 is 10 and 101 | 10 is 111

* **Clear ith bit**

a=a&~(1<<i);//clear ith bit

for eg 111 then 1<<1 is 10 and ~(10) is 01 and 111 & 01 is 101

* **Toggling the ith bit**

a=a^(1<<1);//toggling the ith bit

for eg 101 then 1<<1 is 10 and 101 ^ 10 is 111

* **Remove last set bit**

a=a&a-1;//remove last set bit

* **check if no is power of 2 or not**

int no=33;

if(no&(no-1)){

cout<<"\nfalse:";

}

else{

cout<<"\ntrue:";

}