**BITWISE OPERATOR**

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int main ()

{

  int a=3;   //0000........0011

   int b=6;   //0000........0110

  // bitwise operator: in c/c++ we have 6 bitwise operator

  //1.AND bitwise opertor:it does AND on every bit of two numbers.the result of AND is 1(set) only if both bits are 1

  cout<<(a&b)<<endl;

  //2. OR bitwise opertor:it does OR on every bit of two number.the result is 1(set) if any of the two bits is 1;

  cout<<(a||b)<<endl;

  //3.XOR bitwise operator:it does XOR on every bit of two number.the results is 1(set) if bot bit are different

  cout<<(a^b)<<endl;

  //4.LEFT SHIFT:it takes two parameter as a input first a number whose binary representation is to be shifted second number tells how many time two be shifted

  cout<<(a<<1)<<endl; //shift left 1 times

  cout<<(a<<2)<<endl; //shift left 2 times

  // x left shift y means  x\*2^y

  //5.RIGHT SHIFT: it is opposite of left shift

  cout<<(b>>1)<<endl;

  cout<<(b>>2)<<endl;

  //x right shift y means floor value(x/(2^y))

  //BITWISE NOT: it takes number and inverts all bits of it. take care of signed and unsigned number it might genrate different answer

  cout<<(~a)<<endl;

  cout<<(~b)<<endl;

  return 0;

}

/\*facts:

1.the left shift and right shift shoul not be used for negative number.it results in undefined behavipur in c/c++

2.the bitwise XOR is the most important and useful operator from a technical interview perspective.

3.the AND operator can be used to quickly checked if a number is odd or even (x &1)?cout<<"odd":cout<<"even"