```
#include <stdio.h>
void delay(int n)
{
   for(int i=0;i<n; i++)
      for(int j=0;j<n; j++)
         ;    // no work during this moment (                  // (15K * 15K)
}
int main()
{
   for(int i=0; i<5; i++)
   {
      printf("Hello World\n");
      delay(15000);
   }
   return 0;
}
```

```
//bitwise operator

// arithmatic op: + - * / %   2+3 => 5   4/2= 2

// logical op:   2 && 3 => 1   2 && 0 => 0

// Bitwise op:   works on bits   ( & bitwise AND, | bitwise OR,  ^ EXOR,  ~ NOT,  >> RIGHT SHIFT, << LEFT SHIFT)

//  a= 2   =>    0000 0000 0000 0010
//  b= 3   =>    0000 0000 0000 0011
// c= a & b; =>  0000 0000 0000 0010  (o/p will b 0 if any one of input is 0)
// output (c) = 2

//  a= 2   =>    0000 0000 0000 0010
//  b=3   =>     0000 0000 0000 0011
// c= a | b; =>  0000 0000 0000 0011  (o/p will b 1 if any one of input is 1)
// output (c) = 3

//  a= 2   =>    0000 0000 0000 0010
//  b=3   =>     0000 0000 0000 0011
// c= a ^ b; =>  0000 0000 0000 0001  (o/p will b 1 if both of the inputs are different)
// 0 0   0
// 0 1   1
// 1 0   1
// 1 1   0
// output (c) = 1

//  << left shift
//  >> right shift
// ~ one's complement   a= ~a;    1111 1111 1111 1101

#include <stdio.h>
int main()
{
   int a=5, b;
   printf("bitwise operator\n");
   b=a&5;                       //0000 0000 0000 0101
   printf("Value of b is %d\n",b); //0000 0000 0000 0101
   b=a|5;
   printf("Value of b is %d\n",b);
   b=~a;
   printf("Value of b is %d\n",b);
   return 0;
}


// 3 questions may arise
// NEED of bitwise operators =>  1. prog will be compact, 2. faster
// Ans: bit manipulations

// Q1: Set/reset bit (Conversion to 1 or 0)

   int a =39;  // 0000 0000 0010 0111
  // Q: check the 6th bit of your number. Is it 0 or 1 ??
```

```
// ans:
   int b=1; // 0000 0000 0000 0001
   b=b<<5;  // 0000 0000 0010 0000                    0110
   c=a&b;   // 0000 0000 0010 0000  non0 => 6th bit is SET //   0001    0000   zero  1st bit is 0
   c?printf("6th bit is 1"):printf("6th bit is 0");       //    0010    0010   non 0 2nd bit is 1
                                          //    0100    0100   non 0 3rd bit is 1
                                          //    1000    0000   zero  4th bit is 0
scanf(a);
scanf(pos);
b=1<<(pos-1);
a&b ? printf("pos bit is 1") : printf("pos bit is 0");


// Toggle bits
 int a= 5;  // 0000 0000 0000 0101
         // 1111 1111 1111 1010

// bitwise and operator is used for checking
// bitwise or operator is used to a SET a particular bit
// bitwise xor is used to toggle a particular bit

// Q2: Check bit
#include <stdio.h>
int main()
{
   unsigned int a=15;
   int i, mask, rslt;
   printf("bitwise operator\n");
   printf("In which index position, you want to check? (please enter within range 0 to 15)\t");
   scanf("%d", i);

   actual_pos= 1<<i;      //set index checker
   rslt=num&actual_pos;        //find
   rslt?printf(" 1"):printf(" 0");   //show
   return 0;
}



// ShowBits

#include <stdio.h>
int main()
{
   unsigned int a=15;
   int i, mask, rslt;
   printf("bitwise operator\n");
   for(i=15; i>=0; i--)
   {
      mask=1<<i;
      rslt=a&mask;
      rslt?printf(" 1"):printf(" 0");
   }
   return 0;
}
```

```
//switch case || alternative of nested if-else
#include <stdio.h>
int main()
{
   float v=3.5;    // v= 3.499999   or v= 3.500001  or v= 3.499998
   if(v==3.5)
     printf("Hello");
   else
     printf("Hi");
   return 0;
}
```

Output: "hi"

```
#include <stdio.h>
int main()
{
   float v=3.5f;    // v= 3.5
   if(v==3.5)
     printf("Hello");
   else
     printf("Hi");
   return 0;
}
```

```
#include <stdio.h>
int main()
{
   unsigned int a=32;
   int i, mask, rslt;
   printf("use bitwise operator\n");
   for(i=15; i>=0; i--)       // 16 bits(0-15)
   {
      mask=1<<i;
      rslt=a&mask;
      rslt?printf(" 1"):printf(" 0");   //using conditional operator
   }
   return 0;
}

// rslt?printf(" 1"):printf(" 0");   //using conditional operator
/*
//as same as
if(rslt)
  printf(" 1")
else
  printf(" 0");

// as same as
if(rslt==1)
  printf(" 1")
else
  printf(" 0");
*/



//i=15
// 32 means a=   0000 0000 0010 0000
//line 9: mask=  1000 0000 0000 0000     <=   0000 0000 0000 0001  << 15
//line 10: rslt= 0000 0000 0000 0000  => 0 => printf(" 0")

//i=14
// 32 means a=   0000 0000 0010 0000
//line 9: mask=  0100 0000 0000 0000     <=   0000 0000 0000 0001  << 14
//line 10: rslt= 0000 0000 0000 0000  => 0 => printf(" 0")

// ...

//i=5
// 32 means a=   0000 0000 0010 0000
//line 9: mask=  0000 0000 0010 0000     <=   0000 0000 0000 0001  << 5
//line 10: rslt= 0000 0000 0010 0000  => non zero => printf(" 1")

//....
```

```
//switch case|| alternative of nested if-else
#include<stdio.h>
int main()
{
flote v=3.5; //v= 3.499999 or v= 3.500001 or v= 3.499998
if(v==3.5)
printf("hello");
else
printf("hi");
return 0;
}
```

Output: "hi"

```
#include<stdio.h>
int main()
{
flote v=3.5f; //v= 3.5
if(v==3.5)
printf("hello");
else
printf("hi");
return 0;
}
```

Output: "hello"