**Bitwise Operators**

**---------------------------------------------------------------------------------------------------------------**

1**.**  Write a program for the following one.

a) Set a bit b) Clear a bit c) Toggle a bit

(Logic:

1) Set a bit: num=num|(1<<pos);

2) Clear a bit: num=num&~(1<<pos);

3) Complement a bit: num=num^(1<<pos);

)

2. WAP to find the given number is even or odd using bitwise operators.

(Logic:

int x;

if(x&1)

printf(“Odd number”)

else

printf(“Even number”)

)

3. WAP to find the given number is +ve or -ve using bitwise operators.

#include<stdio.h>

main()

{

int x, len;

printf("Enter the number: \n");

scanf("%d", &x);

len=(sizeof(x)\*8)-2;

if(x>>len)

{

printf("Negative\n");

}

else

{

printf("Positive\n");

}

}

#include<stdio.h>

int main()

{

int n, pos=31;

printf("Enter any number\n");

scanf("%d", &n);

if(n&1<<pos)

printf("Negative\n");

else

printf("Positive\n");

}

4. WAP to swap two numbers using bitwise operators.

(Logic: i=1010

j=1111

i^j=(1010^1111)

i=i^j //(i^j=0101)

j=i^j //(i^j=1010)

i=j^i //(1010^0101=1111)

print i and j;)

5. WAP to find the given number is power of 2 or not.

#include<stdio.h>

int main()

{

int k;

printf("Enter the value of k\n");

scanf("%d", &k);

if((k&(k-1))==0)

{

printf("Entered value is power of 2\n");

}

else

{

printf("Entered value is not power of 2\n");

}

}

6. WAP to find the given number is divisble by 8 or not using bitwise operators.

#include<stdio.h>

int main()

{

int x;

printf("Enter the value of x\n");

scanf("%d", &x);

if((x&7)==0)

{

printf("The number %d is divisble by 8\n", x);

}

else

{

printf("The number %d is not divisible by 8\n", x);

}

}

7. Write a program to rotate the bits. Input the no.of rotations, at runtime.

Ex : binary : 10000000000000000000000000001011

rotations : suppose 3 times right, then

result : 01110000000000000000000000000001

binary : 10000000000000000000000000001011

rotations : suppose 4 times left, then

result : 00000000000000000000000010111000

#include<stdio.h>

#define INT\_BITS 32

/\*Function to left rotate n by d bits\*/

int leftRotate(int n, unsigned int d)

{

/\* In n<<d, last d bits are 0. To put first 3 bits of n at

last, do bitwise or of n<<d with n >>(INT\_BITS - d) \*/

return (n << d)|(n >> (INT\_BITS - d));

}

/\*Function to right rotate n by d bits\*/

int rightRotate(int n, unsigned int d)

{

/\* In n>>d, first d bits are 0. To put last 3 bits of at

first, do bitwise or of n>>d with n <<(INT\_BITS - d) \*/

return (n >> d)|(n << (INT\_BITS - d));

}

/\* Driver program to test above functions \*/

int main()

{

int n, d;

printf("Enter the number\n");

scanf("%d", &n);

printf("Enter the number of count for shifting the number\n");

scanf("%d", &d);

printf("Left Rotation of %d by %d is ", n, d);

printf("%d\n", leftRotate(n, d));

printf("\nRight Rotation of %d by %d is ", n, d);

printf("%d\n", rightRotate(n, d));

getchar();

}

8. Convert the characters Upper to Lower and Lower to Upper using bitwise

operators.

#include<stdio.h>

int main()

{

char ch;

printf("Enter the character\n");

scanf("%c",&ch);

ch=ch^32;

printf("%c\n", ch);

}

9. Write a program to reverse the bits of a given number.

Note : not just reverse printing.

#include<stdio.h>

int main()

{

int num, m , n, i, j;

printf("Enter the number\n");

scanf("%d", &num);

printf("Before reversal of the bits\n");

for(i=31; i>=0; i--)

{

printf("%d", num>>i&1);

}

printf("\n");

for(i=31, j=0; j<i; i--, j++)

{

m=num>>i&1;

n=num>>j&1;

if(m!=n)

{

num=num^1<<i;

num=num^1<<j;

}

}

printf("After reversal of the bits the result is \n");

for(i=31; i>=0; i--)

{

printf("%d", num>>i&1);

}

printf("\n");

}

10. Write a one line code to compare two numbers using bitwise operators.

z=x^y;

#include<stdio.h>

int main()

{

int x, y, z=0;

printf("Enter the value of 1st number\n");

scanf("%d", &x);

printf("Enter the value of 2nd number\n");

scanf("%d", &y);

z=x^y;

if(z)

{

printf("Both numbers are not equal\n");

}

else

{

printf("Both numbers are equal\n");

}

if(x<y)

{

printf("x is less than y\n");

}

else

printf("x is greater than y\n");

}

11. Write a program to print float binay formation using char \*ptr.

#include<stdio.h>

int main()

{

int i;

float f=3.5;

char\* cp;

cp=&f;

for(i=31; i>=0; i--)

{

printf("%d", \*cp>>i&1);

cp=cp-1;

}

printf("\n");

}

12. Write a program to swap the adjucent bytes of a given 4-digit hexadecimal

number.

Ex : given number = 0x1234;

after swap : 0x3412;

#include<stdio.h>

int main()

{

short int n;

printf("Enter any hexadecimal number\n");

scanf("%hx", &n);

printf("Entered number is 0x%hx\n", n);

n=n>>8|n<<8;

printf("After swap number is 0x%hx\n", n);

}

13. Write a program to delete no.of bits from perticular position in a given number.

Input the no.of bits, at runtime.

Ex: Suppose num = 100;

It's Binaray is 00000000000000000000000001100100

delete 2 bits from 4th position

then result is 00000000000000000000000000011100

#include<stdio.h>

int main()

{

int n, i, k, l;

printf("Enter any number\n");

scanf("%d", &n);

printf("The number is :\n");

for(i=31; i>=0; i--)

{

printf("%d", n>>i&1);

}

printf("\n");

printf("Enter the position which you want to delete\n");

scanf("%d", &k);

printf("Enter the position number which you want to delete\n");

scanf("%d", &l);

for(i=31; i>=0; i--)

{

if(i!=k)

{

printf("%d", n>>i&1);

}

else

{

i=i-l;

printf("%d", n>>i&1);

}

}

printf("\n");

}

14. Write a macro for swapping first and last nibbles in a given integer.

Ex: Suppose num = 10

It's Binary is 0000000000000000000000000001010

After swap 1010000000000000000000000000000

#include<stdio.h>

void main()

{

int n, i;

printf("Enter the number :");

scanf("%d",&n);

printf("Binary:");

for(i=31; i>=0; i--)

printf("%d", n>>i&1);

printf("\n");

n=n>>28|n<<28;

printf("After swap binary:");

for(i=31; i>=0; i--)

printf("%d", n>>i&1);

printf("\n");

}

15. Write a logic to extract P bits from Position N in an integer M

#include<stdio.h>

int extract\_bits(int, int);

int main()

{

int n, i, a, b, c, result;

printf("Enter any number: \n");

scanf("%d", &n);

printf("The binary of the above number is:\n");

for(i=31; i>=0; i--)

printf("%d", n>>i&1);

printf("\n");

printf("Enter the starting value from where you want to extract the digits:\n");

scanf("%d", &a);

printf("Enter the ending value of the digit till you want to extract the digits:\n");

scanf("%d", &b);

c=extract\_bits(a, b);

result=c&n;

for(i=31; i>=0; i--)

printf("%d", result>>i&1);

printf("\n");

}

int extract\_bits(int a, int b)

{

int d=0, i;

for(i=a; i<=b; i++)

{

d=d|(1<<i);

}

return d;

}

16. Write a macro to clear a bit at the position N in an integer M.

#include<stdio.h>

#define clear num=num&~(1<<pos)

int main()

{

int num, pos;

printf("Enter the number: \n");

scanf("%d", &num);

printf("Enter the position:\n");

scanf("%d", &pos);

printf("The resultant number after clearing the bits is:%d\n", clear);

}

17. There are 48 bits are stored in an array of character buffer and store them into 2

integer variables.

-------------------------------------------------------- END --------------------------------------------------------

Dear Students, if any mistakes found, Kindly inform to me.

A.Tandava Ramakrishna

Email: ramakrishna@vectorindia.org