1. Program to reverse bits of an integer number

#include <stdio.h>

unsigned int reverse(unsigned int num)

{

unsigned int mask1=0x80000000,mask2=1,reversenum=0;

int bits=0;

bits=sizeof(unsigned int)\*8;

for (int i=0;i<bits;i++,mask1>>=1)

{

if(num & mask1)

{

reversenum|=mask2;

mask2<<=1;

}

else

{

mask2<<=1;

}

}

return reversenum;

}

int main()

{

unsigned int num,reversenum;

printf("Please enter an integer number:");

scanf("%u",&num);

reversenum=reverse(num);

printf("The reverse number is %u\n",reversenum);

return 0;

}

Output:

**Please enter an integer number:**2

**The reverse number is 1073741824**

**Please enter an integer number:**1

**The reverse number is 2147483648**

**Please enter an integer number:**12345

**The reverse number is 2618032128**

1. Program to count number of 1’s in a number

#include <stdio.h>

unsigned int count1bits(unsigned int num)

{

unsigned int mask=1;

int bits=0,all1bits=0;

bits=sizeof(unsigned int)\*8;

for (int i=0;i<bits;i++,mask<<=1)

{

if(num & mask)

all1bits++;

}

return all1bits;

}

int main()

{

unsigned int num,all1bits;

printf("Please enter an integer number:");

scanf("%u",&num);

all1bits=count1bits(num);

printf("The number of 1 bit is %u\n",all1bits);

return 0;

}

Output:

**Please enter an integer number:**15

**The number of 1 bit is 4**

**Please enter an integer number:**65535

**The number of 1 bit is 16**

1. Program to swap nibbles of a byte.

#include <stdio.h>

unsigned int SwapNibbles(unsigned int num)

{

unsigned int numbyte1,numbyte2,swapnum;

numbyte1=num>>4;

numbyte2=num<<4;

swapnum=numbyte1+numbyte2;

return swapnum;

}

int main()

{

unsigned int num,swapnum;

printf("Please enter an integer number:");

scanf("%u",&num);

swapnum=SwapNibbles(num);

printf("The integer number after swapping nibbles is %u\n",swapnum);

return 0;

}

Output:

**Please enter an integer number:**3

**The integer number after swapping nibbles is 48**

**Please enter an integer number:**1

**The integer number after swapping nibbles is 16**

1. Program to get minimum number of bits to store an integer number

#include <stdio.h>

unsigned int CountMinBits(unsigned int num)

{

unsigned int mask=0x80000000;

int bits=0,minbits=0;

bits=sizeof(unsigned int)\*8;

for (int i=0;i<bits;i++,mask>>=1)

{

if(num & mask)

{

minbits=bits-i;

break;

}

}

return minbits;

}

int main()

{

unsigned int num,minbits;

printf("Please enter an integer number:");

scanf("%u",&num);

minbits=CountMinBits(num);

printf("The minmum number of bits to store this integer is %u\n",minbits);

return 0;

}

Output:

**Please enter an integer number:**65535

**The minmum number of bits to store this integer is 16**

**Please enter an integer number:**8

**The minmum number of bits to store this integer is 4**

1. Program to demonstrate example of left shift (<<) operator, >>, &, ~ , ^ , | bitwise operators.

#include <stdio.h>

int main()

{

    unsigned int num1,num2;

    unsigned int rsnum=0,lsnum=0,andnum=0,notnum=0,xornum=0,ornum=0;

    printf("Please enter an integer number 1:");

    scanf("%u",&num1);

    printf("Please enter an integer number 2:");

    scanf("%u",&num2);

    rsnum=num1>>1;

    lsnum=num2<<1;

    andnum=num1&num2;

    notnum=~num1;

    xornum=num1^num2;

    ornum=num1|num2;

    printf("num1 0x%x right shfit 1 bit is 0x%x\n",num1,rsnum);

    printf("num2 0x%x left shfit 1 bit is 0x%x\n",num2,lsnum);

    printf("num1 0x%x and num2 0x%x is 0x%x\n",num1,num2,andnum);

    printf("not num1 0x%x is 0x%x\n",num1,notnum);

    printf("num1 0x%x xor num2 0x%x is 0x%x\n",num1,num2,xornum);

    printf("num1 0x%x or num2 0x%x is 0x%x\n",num1,num2,ornum);

    return 0;

}

Output:

**Please enter an integer number 1:**128

**Please enter an integer number 2:**255

**num1 0x80 right shfit 1 bit is 0x40**

**num2 0xff left shfit 1 bit is 0x1fe**

**num1 0x80 and num2 0xff is 0x80**

**not num1 0x80 is 0xffffff7f**

**num1 0x80 xor num2 0xff is 0x7f**

**num1 0x80 or num2 0xff is 0xff**