Table of Contents

WAP to check status of a given bit of a Number. 2

Write a program for the following one... 3

WAP to find the given number is even or odd using bitwise operators. 5

WAP to find the given number is +ve or -ve using bitwise operators. 6

WAP to swap two numbers using bitwise operators. 7

WAP to find the given number is power of 2 or not. 8

WAP to find the given number is divisible by 8 or not using bitwise operators. 9

Write a program to rotate the bits. Input the no of rotations, at runtime. 10

Convert the characters Upper to Lower & Lower to Upper using bitwise operators. 11

Write a program to reverse the bits of a given number.. 12

Write a one line code to compare two numbers using bitwise operators. 14

Write a program to print float binary formation using char *ptr. 15

Write a program to swap the adjacent bytes of a given 4-digit hex number. 17

Write a program to delete no of bits from particular position in a given number. 19

Write a macro for swapping first and last nibbles in a given integer. 19

Write a logic to extract P bits from Posion N in an integer M.................................. 19

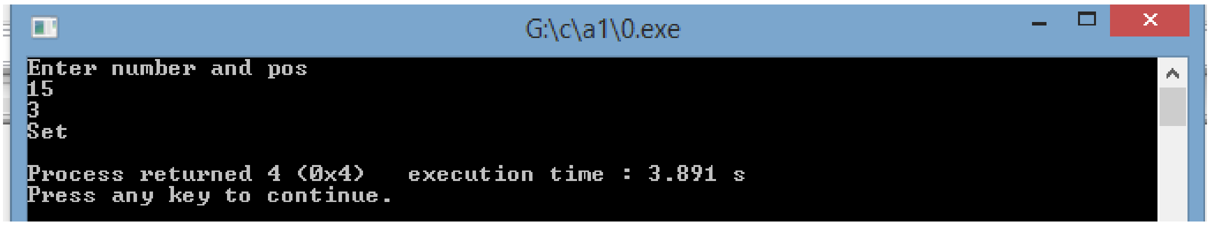
Write a macro to clear a bit at the position N in an integer M. 19

There are 48 bits are stored in an array of character buffer and store them into 2 integer variables. 19

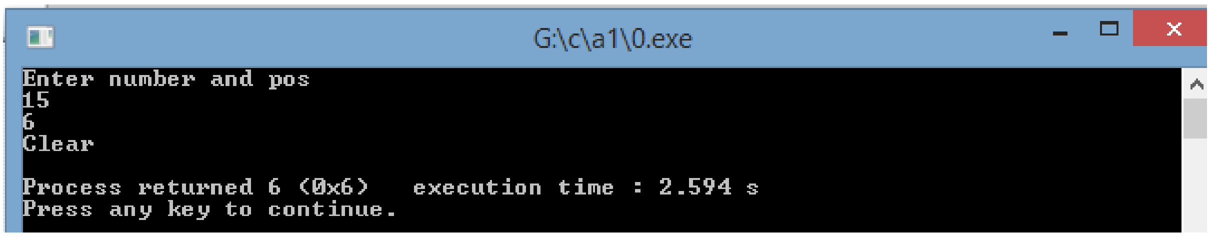
BITWISE OPERATORS

WAP to check status of a given bit of a Number.

```
#include<stdio.h>
main()
{
        int num,pos,r;

        printf("Enter number and pos\n");
        scanf("%d %d",&num,&pos);

        (num >> pos & 1) ? printf("Set\n"):printf("Clear\n");
        //(num & 1 << pos) > 0 printf("Set\n"):printf("Clear\n");
}
```



```
Enter number and pos
15
3
Set

Process returned 4 (0x4)   execution time : 3.891 s
Press any key to continue.
```



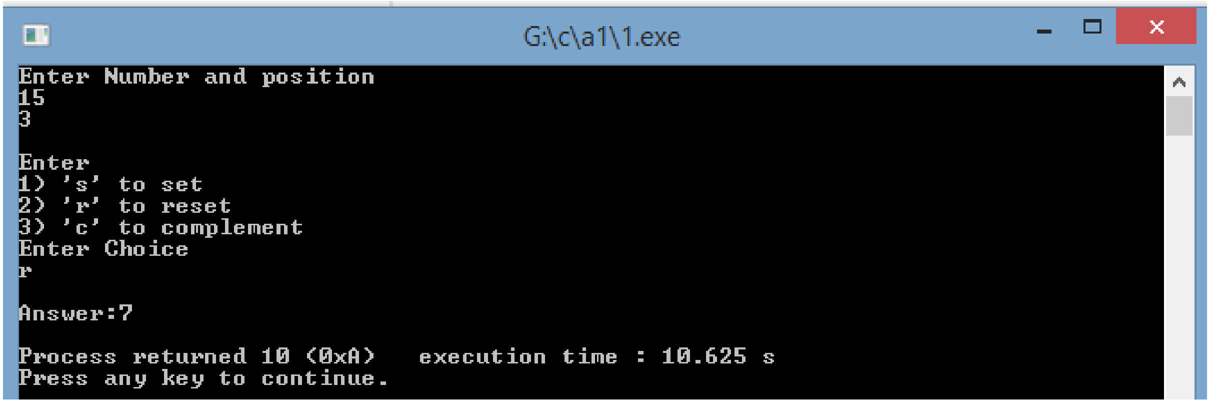
```
Enter number and pos
15
6
Clear

Process returned 6 (0x6)   execution time : 2.594 s
Press any key to continue.
```

Write a program for the following one.

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

```
#include<stdio.h>

main()
{
        int num,pos,r;
        char op;
        printf("Enter Number and position\n");
        scanf("%d %d",&num,&pos);

        printf("\nEnter\n1) 's' to set\n2) 'r' to reset\n3) 'c' to complement\n");
        printf("Enter Choice\n");
        scanf(" %c",&op);

        switch(op)
        {
                case 's':
                num = num | 1 << pos;
                printf("\nAnswer:%d\n",num);
                break;

                case 'r':
```

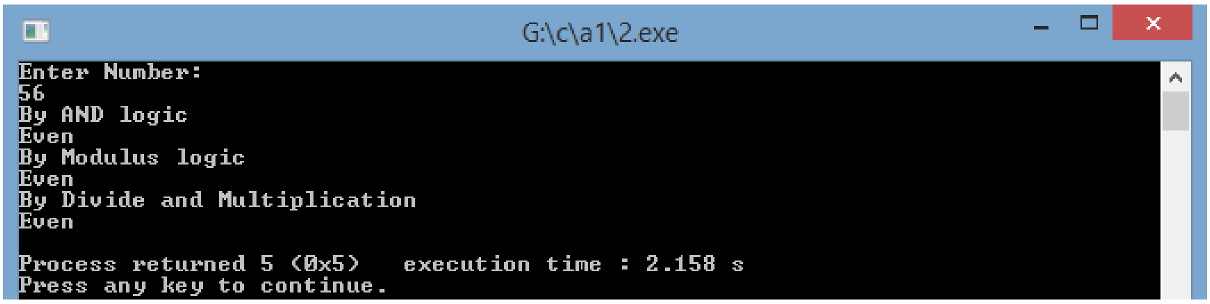
```
                num = num & ~(1 << pos);

                printf("\nAnswer:%d\n",num);

                break;


                case 'c':

                num = num ^ (1 << pos);

                printf("\nAnswer:%d\n",num);

                break;


                default:

                printf("\nInvalid Chice\n");

        }

}
```



```
Enter Number and position
15
3

Enter
1) 's' to set
2) 'r' to reset
3) 'c' to complement
Enter Choice
r

Answer:7

Process returned 10 (0xA)   execution time : 10.625 s
Press any key to continue.
```

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

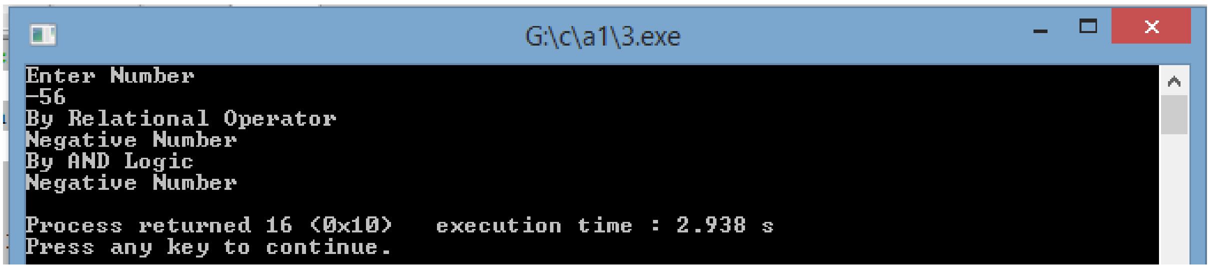
```
#include<stdio.h>
main()
{
        int num;

        printf("Enter Number:\n");
        scanf("%d",&num);

        printf("By AND logic\n");
        (num & 1)?printf("Odd\n"):printf("Even\n");

        printf("By Modulus logic\n");
        (num % 2)?printf("Odd\n"):printf("Even\n");

        printf("By Divide and Multiplication\n");
        ((num/2) * 2 == num) ? printf("Even\n"):printf("Odd\n");;
}
```



```
Enter Number:
56
By AND logic
Even
By Modulus logic
Even
By Divide and Multiplication
Even

Process returned 5 (0x5)   execution time : 2.158 s
Press any key to continue.
```

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

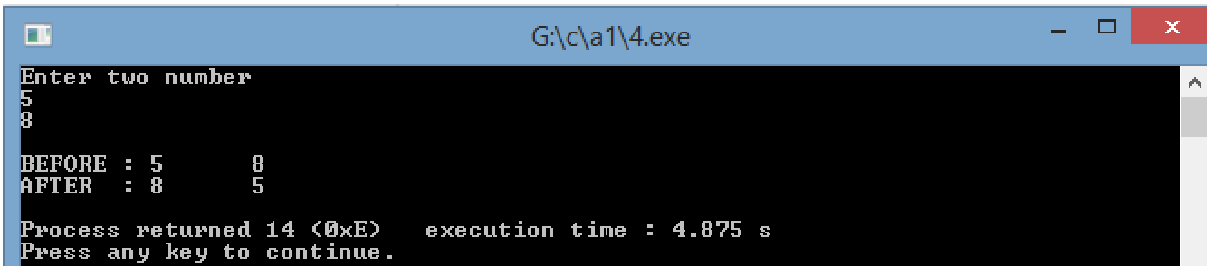
```
#include<stdio.h>


main()
{
        int num;
        printf("Enter Number\n");
        scanf("%d",&num);

        printf("By Relational Operator\n");
        if(num < 0)
                printf("Negative Number\n");
        else
                printf("Positive Number");
        printf("By AND Logic\n");
        (num & 1 << sizeof(int)*8-1) ? printf("Negative
Number\n"):printf("Positive Number\n");
}
```



```
Enter Number
-56
By Relational Operator
Negative Number
By AND Logic
Negative Number

Process returned 16 (0x10)   execution time : 2.938 s
Press any key to continue.
```

WAP to swap two numbers using bitwise operators.

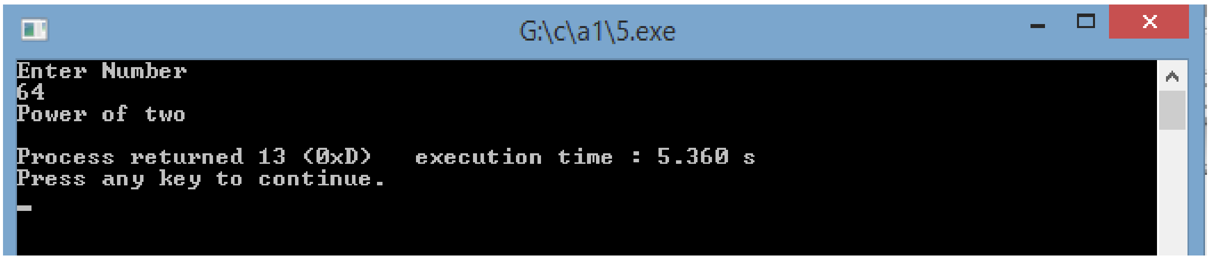
```
#include<stdio.h>

main()

{

        int num1,num2;

        printf("Enter two number\n");

        scanf("%d %d",&num1,&num2);

        printf("\nBEFORE : %d \t%d\n",num1,num2);

        num1 = num1 ^ num2;

        num2 = num1 ^ num2;

        num1 = num1 ^ num2;


        //num2 = num1+num2 - (num1=num2);

        //num2 = num1*num2 / (num1=num2);

        printf("AFTER  : %d \t%d\n",num1,num2);


}
```



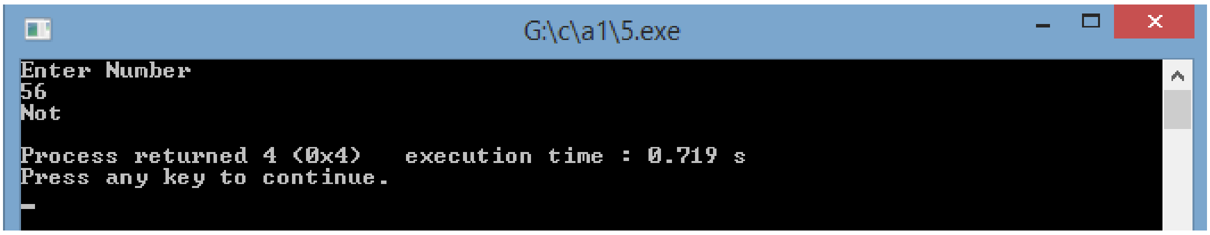
```
Enter two number
5
8

BEFORE : 5       8
AFTER  : 8       5

Process returned 14 (0xE)   execution time : 4.875 s
Press any key to continue.
```

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

```
#include<stdio.h>


main()

{

        int num;

        printf("Enter Number\n");

        scanf("%d",&num);


        (num & num - 1)? printf("Not\n"):printf("Power of two\n");

}
```



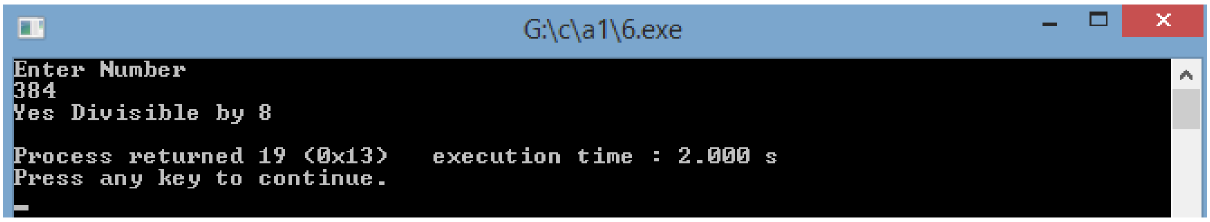
```
Enter Number
64
Power of two

Process returned 13 (0xD)   execution time : 5.360 s
Press any key to continue.
```



```
Enter Number
56
Not

Process returned 4 (0x4)   execution time : 0.719 s
Press any key to continue.
```

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

```
#include<stdio.h>


main()
{
        int num;
        printf("Enter Number\n");
        scanf("%d",&num);

        num & 7 ? printf("No\n"):printf("Yes Divisible by 8\n");
}
```



```
Enter Number
384
Yes Divisible by 8

Process returned 19 (0x13)   execution time : 2.000 s
Press any key to continue.
```

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

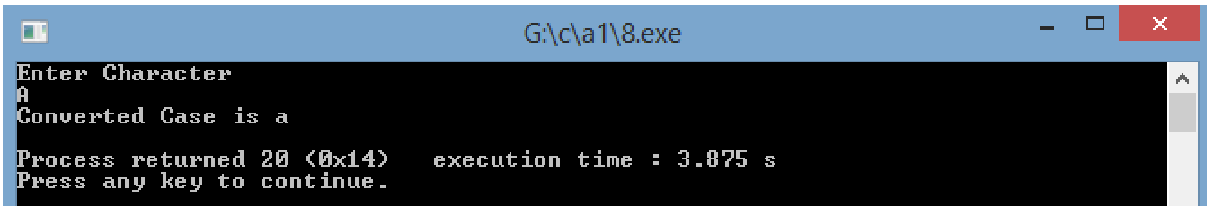
binary : 10000000000000000000000000001011

rotations : suppose 4 times left, then

result : 00000000000000000000000010111000

Convert the characters Upper to Lower & Lower to Upper using bitwise operators.

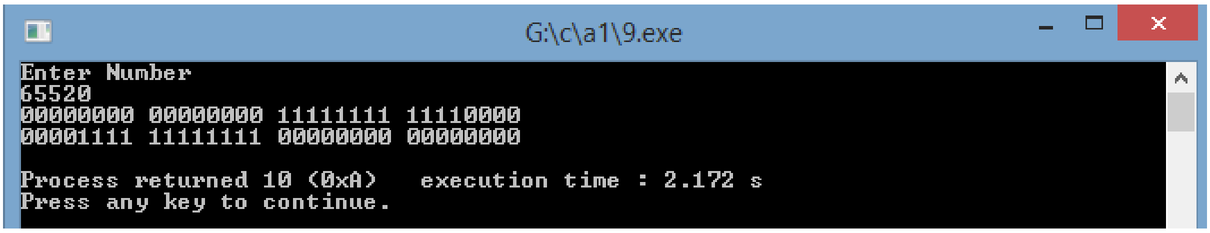
```
#include<stdio.h>


main()
{
        char n;


        printf("Enter Character\n");
        scanf("%c",&n);


        n = n ^ 32;


        //n = n ^ (1 << 5);
        //n = n ^ ' ';


        printf("Converted Case is %c\n",n);
}
```



```
Enter Character
A
Converted Case is a

Process returned 20 (0x14)   execution time : 3.875 s
Press any key to continue.
```

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

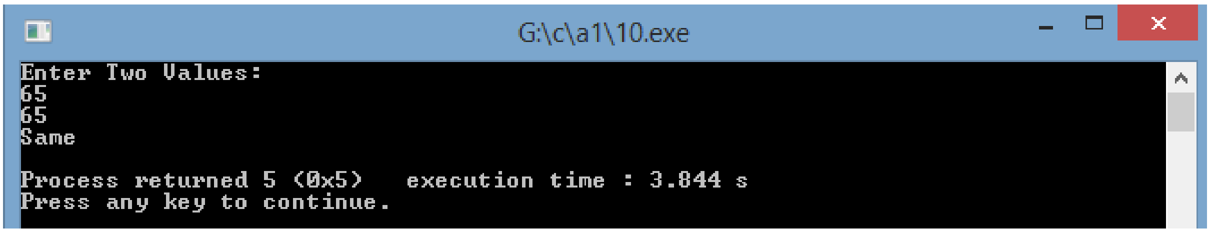
```
#include<stdio.h>


main()
{
        int num,i,j;


        printf("Enter Number\n");
        scanf("%d",&num);


        for(i=sizeof(int)*8-1;i>=0;i--)
        {
                printf("%d",num >> i & 1);
                if (i % 8 == 0)
                        printf(" ");
        }
        printf("\n");





        for(i=0,j=sizeof(int)*8-1;i<(sizeof(int)*8)/2;i++,j--)
        {


                if( (num >> i & 1) != (num >> j & 1) )
                {

                        num = num ^ 1 << i;

                        num = num ^ 1 << j;

                }

        }



        for(i=sizeof(int)*8-1;i>=0;i--)

        {

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

                if (i % 8 == 0)

                        printf(" ");

        }

        printf("\n");

}
```



```
Enter Number
65520
00000000 00000000 11111111 11110000
00001111 11111111 00000000 00000000

Process returned 10 (0xA)   execution time : 2.172 s
Press any key to continue.
```

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

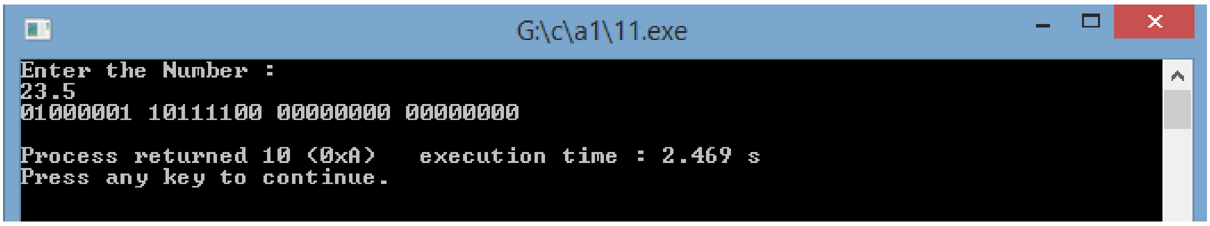
```
#include<stdio.h>


main()
{
        int n1,n2;


        printf("Enter Two Values:\n");
        scanf("%d %d",&n1,&n2);



        n1 ^ n2 ? printf("Different\n"):printf("Same\n");
}
```



```
Enter Two Values:
65
65
Same

Process returned 5 (0x5)   execution time : 3.844 s
Press any key to continue.
```

Write a program to print float binary formation using char *ptr.

```
#include<stdio.h>
main()
{
        float num;
        char *c=&num;
        int i,j;

        printf("Enter the Number : \n");
        scanf("%f",&num);

        c=c+3;
        for(i=0;i<4;i++)
        {
                for(j=7;j>=0;j--)
                {
                        printf("%d",(*c) >> j & 1);
                }
        printf(" ");
        c=c-1;
        }
        printf("\n");

}
```



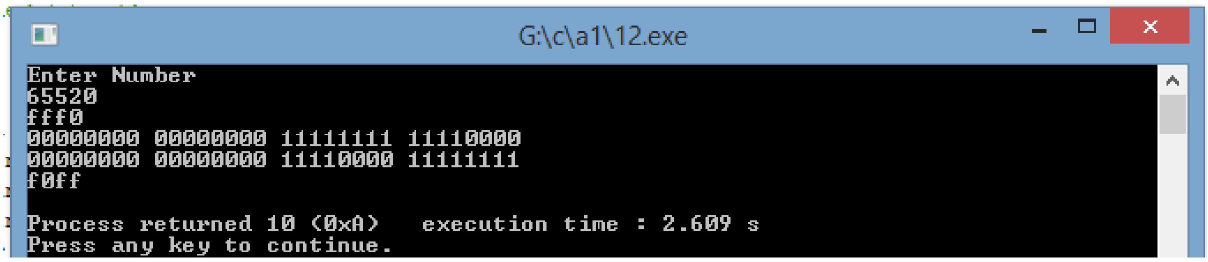
```
Enter the Number :
23.5
01000001 10111100 00000000 00000000

Process returned 10 (0xA)   execution time : 2.469 s
Press any key to continue.
```

Write a program to swap the adjacent bytes of a given 4-digit hex number.

Ex : given number = 0x1234;

after swap : 0x3412;

```
#include<stdio.h>

main()

{

        int i,j,k,num,m,n;

        printf("Enter Number\n");

        scanf("%d",&num);

        printf("%x\n",num);

        for(i=sizeof(int)*8-1;i>=0;i--)

        {

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

                        if(i % 8 == 0)

                        printf(" ");

        }

        printf("\n");


        for(i=0,j=8;i<=7;i++,j++)

        {


                if( (num >> i & 1) != (num >> j & 1))
        {
            num = num ^ 1 << i;
            num = num ^ 1 << j;
        }
    }

    //num = (num & 0xff) << 8 | (num & 0xff00) >> 8;
    for(i=sizeof(int)*8-1;i>=0;i--)
    {
        printf("%d",num >> i & 1);
            if(i % 8 == 0)
            printf(" ");
    }

printf("\n%x",num);
printf("\n");



}
```



```
Enter Number
65520
fff0
00000000 00000000 11111111 11110000
00000000 00000000 11110000 11111111
f0ff

Process returned 10 (0xA)   execution time : 2.609 s
Press any key to continue.
```

Write a program to delete no of bits from particular 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

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

Write a logic to extract P bits from Posion N in an integer M

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

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