**A31 - WAP to count number of set bits in a given number and print parity**

/\*

Name : Nestin Gregorios Sunny

Date : 18.06.2025

Description :

Program to count set bits of a number and print parity

Sample Input :

Enter the number : 7

Sample Output :

Number of set bits = 3

Bit parity is Odd

\*/

#include<stdio.h>

int main()

{

int num, i;

int count = 0;

printf("Enter the number :");

scanf("%d", &num);

for(i = 0; i < 31; i++)

{

if(num & 1 << i)

{

count++;

}

}

printf("Number of set bits = %d\n", count);

if(count & 1)

{

printf("Bit parity is Odd\n");

}

else

{

printf("Bit parity is Even\n");

}

return 0;

}

**A32 - WAP to check N th bit is set or not, If yes, clear the M th bit**

/\*

Name : Nestin Gregorios Sunny

Date : 18.06.2025

Description :

Program to check nth bit of a number is set or not.If set, clear mth bit and print updated number.

Sample Input :

Enter the number: 19

Enter 'N': 1

Enter 'M': 4

Sample Output :

Updated value of num is 3

\*/

#include<stdio.h>

int main()

{

int num, n, m;

printf("Enter the number : ");

scanf("%d", &num);

printf("Enter 'N' : ");

scanf("%d", &n);

printf("Enter 'M' : ");

scanf("%d", &m);

if((num & (1 << n)) != 0)

{

num = num & ~(1 << m);

printf("Updated value of num is %d\n",num);

}

else

{

printf("Updated value of num is %d\n",num);

}

return 0;

}

**A33 - WAP to get 'n' bits of a given number**

/\*

Name : Nestin Gregorios Sunny

Date : 20.06.2025

Description :

Get 'n' bits of a given number

Sample Input :

Enter the number: 10

Enter number of bits: 3

Sample Output :

Result = 3

\*/

#include <stdio.h>

int get\_nbits(int, int);

int main()

{

int num, n, res = 0;

printf("Enter num and n:");

scanf("%d%d", &num, &n);

res = get\_nbits(num, n);

printf("Result = %d\n", res);

}

int get\_nbits(int num, int n)

{

int i, get;

for (i = 0; i < 31; i++)

{

get = (num & ((1 << n) - 1));

}

return get;

}

**A34 - WAP to replace 'n' bits of a given number**

/\*

Name : Nestin Gregorios Sunny

Date : 20.06.2025

Description :

Replace 'n' bits of a given number

Sample Input :

Enter the number: 10

Enter number of bits: 3

Enter the value

Sample Output :

Result = 12

\*/

#include <stdio.h>

int replace\_nbits(int, int, int);

int main()

{

int num, n, val, res = 0;

printf("Enter num, n and val:");

scanf("%d%d%d", &num, &n, &val);

res = replace\_nbits(num, n, val);

printf("Result = %d\n", res);

}

int replace\_nbits(int num, int n, int val)

{

// Create a mask to clear n bits

int mask = (1 << n) - 1;

// Clear n bits in num

num &= ~(mask);

// Get n bits from val and shift them to the correct position

val &= mask;

// Replace n bits in num with n bits from val

return num | val;

}

**A35 - WAP to print 'n' bits from LSB of a number**

/\*

Name : Nestin Gregorios Sunny

Date : 20.06.2025

Description :

Print 'n' from LSB of a number

Sample Input :

Enter the number: 10

Enter number of bits: 12

Sample Output :

Binary form of 10: 0 0 0 0 0 0 0 1 0 1 0

\*/

#include <stdio.h>

int print\_bits(int, int);

int main()

{

int num, n;

printf("Enter num, n :\n");

scanf("%d%d", &num, &n);

printf("Binary form of %d:", num);

print\_bits(num, n);

}

int print\_bits(int num, int n)

{

int i;

for(i = n - 1; i >= 0; i--)

{

if(num & (1 << i))

{

printf("1 ");

}

else

{

printf("0 ");

}

}

return 0;

}

**A38 - WAP to get 'n' bits from given position of a number**

/\*

Name : Nestin Gregorios Sunny

Date : 23.06.2025

Description :

Get n bits from position of a number

Sample Input :

Enter the number : 12

Enter number of bits : 3

Enter the pos : 4

Sample Output :

Result = 3

\*/

#include <stdio.h>

int get\_nbits\_from\_pos(int, int, int);

int main()

{

int num, n, pos, res = 0;

printf("Enter num, n and val:");

scanf("%d%d%d", &num, &n, &pos);

res = get\_nbits\_from\_pos(num, n, pos);

printf("Result = %d\n", res);

}

int get\_nbits\_from\_pos(int num, int n, int pos)

{

int i, mask, get;

for(i = 0; i < 31; i++)

{

mask = ((1 << n) - 1) << ((pos - n) + 1);

get = (num & mask) >> ((pos - n) + 1);

}

return get;

}

**A39 - WAP to toggle 'n' bits from given position of a number**

/\*

Name : Nestin Gregorios Sunny

Date : 23.06.2025

Description :

Toggle 'n' bits from the given position of a number

Sample Input :

Enter the number: 10

Enter number of bits: 3

Enter the pos: 5

Sample Output :

Result = 50

\*/

#include <stdio.h>

int toggle\_nbits\_from\_pos(int, int, int);

int main()

{

int num, n, pos, res = 0;

printf("Enter num, n and val:");

scanf("%d%d%d", &num, &n, &pos);

res = toggle\_nbits\_from\_pos(num, n, pos);

printf("Result = %d\n", res);

}

int toggle\_nbits\_from\_pos(int num, int n, int pos)

{

int i, mask, toggle;

for(i = 0; i < 31; i++)

{

mask = ((1 << n) - 1) << ((pos - n) + 1);

toggle = num ^ mask;

}

return toggle;

}

**A36 - WAP to implement Circular left shift**

/\*

Name : Nestin Gregorios Sunny

Date : 24.06.2025

Description :

Implement Circular Left Shift

Sample Input :

Enter num: 12

Enter n : 3

Sample Output :

Result in Binary: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0

\*/

#include <stdio.h>

int circular\_left(int, int);

int print\_bits(int);

int main()

{

int num, n, ret;

printf("Enter the num:");

scanf("%d", &num);

printf("Enter n:");

scanf("%d", &n);

ret = circular\_left(num, n);

printf("\nResult in Binary: ");

print\_bits(ret);

}

int circular\_left(int num, int n)

{

int clear\_mask, get\_mask, replace;

clear\_mask = num << n;

get\_mask = num & (((1 << n) - 1) << (7 - n + 1));

get\_mask = (unsigned)get\_mask >> (7 - n + 1);

replace = clear\_mask | get\_mask;

return replace;

}

int print\_bits(int ret)

{

int i;

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

{

(ret & (1 << i)) ? printf("1 ") : printf("0 ");

}

printf("\n");

return 0;

}

**A40 - WAP to implement Circular right shift**

/\*

Name : Nestin Gregorios Sunny

Date : 24.06.2025

Description :

Implement Circular Right Shift

Sample Input :

Enter num: 12

Enter n : 3

Sample Output :

Result : 10000000 00000000 00000000 00000001

\*/

#include <stdio.h>

int circular\_right(int, int);

int print\_bits(int);

int main()

{

int num, n, ret;

printf("Enter the num:");

scanf("%d", &num);

printf("Enter n:");

scanf("%d", &n);

ret = circular\_right(num, n);

printf("\nResult in Binary: ");

print\_bits(ret);

}

int circular\_right(int num, int n)

{

int clear, get, replace;

clear = (unsigned) num >> n;

get = (unsigned)num << (32 - n);

replace = clear | get;

return replace;

}

int print\_bits(int ret)

{

int i;

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

{

(ret & (1 << i)) ? printf("1 ") : printf("0 ");

}

printf("\n");

return 0;

}

**A37 - WAP to put the (b-a+1) lsb’s of num into val[b:a]**

/\*

Name : Nestin Gregorios Sunny

Date : 25.06.2025

Description :

Write a program to put the (b - a + 1) lsb's of num into value[b : a]

Sample Input :

Enter the value of 'num' : 11

Enter the value of 'a' : 3

Enter the value of 'b' : 5

Enter the value of 'val' : 174

Sample Output :

Result : 158

\*/

#include <stdio.h>

int replace\_nbits\_from\_pos(int, int, int, int);

int main()

{

int num, a, b, val, res = 0;

printf("Enter num, a, b, and val:");

scanf("%d%d%d%d", &num, &a, &b, &val);

res = replace\_nbits\_from\_pos(num, a, b, val);

printf("Result = %d\n", res);

}

int replace\_nbits\_from\_pos(int num, int a, int b, int val)

{

int n, get;

n = b - a + 1;

if (b >= 0 && b <= 31)

{

if (a <= b)

{

val = val & ~((1 << a) - 1 << n);

get = (num & ((1 << n) - 1)) << n;

val = val | get;

return val;

}

}

return num;

}