

1. Write a C program to determine if the least significant bit of a given integer is set (i.e., check if the number is odd).

```
#include<stdio.h>

int main()
{
    int num;

    printf("Enter the number: ");
    scanf("%d",&num);

    if(num & 1)
    {
        printf("lsb is odd");
    }
    else
    {
        printf("lsb is even");
    }

    return 0;
}
```

2. Create a C program that retrieves the value of the nth bit from a given integer.

```
#include<stdio.h>

int main()
{
    int num,n,res;
    printf("Enter the number: \n");
    scanf("%d",&num);
    printf("Enter n: ");
    scanf("%d",&n);
    res=(num>>n)&1;
    printf("%d",res);
    return 0;
}
```

3) Develop a C program that sets the nth bit of a given integer to 1.

```
#include<stdio.h>

int main()
{
    int num,n,res;
    printf("Enter the number: \n");
    scanf("%d",&num);
    printf("Enter n: ");
    scanf("%d",&n);
    res=num|(1<<n);
    printf("%d",res);
    return 0;
}
```

4) Write a C program that clears (sets to 0) the nth bit of a given integer.

```
#include<stdio.h>

int main()
{
    int num,n,res;
    printf("Enter the number: \n");
    scanf("%d",&num);
    printf("Enter n: ");
    scanf("%d",&n);
    res=num&~(1<<n);
    printf("%d",res);
    return 0;
}
```

5) Create a C program that toggles the nth bit of a given integer.

```
#include<stdio.h>

int main()
{
    int num,n,res;

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

    scanf("%d",&num);

    printf("Enter n: ");

    scanf("%d",&n);

    res=num^(1<<n);

    printf("%d",res);

    return 0;
}
```

6) Write a C program that takes an integer input and multiplies it by  $2^n$  using the left shift operator.

```
#include<stdio.h>

int main()
{
    int num,n,res;

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

    scanf("%d",&num);

    printf("Enter n: ");

    scanf("%d",&n);

    res=num<<n;

    printf("%d",res);

    return 0;
}
```

```
}
```

7) Create a C program that counts how many times you can left shift a number before it overflows (exceeds the maximum value for an integer).

```
#include <stdio.h>

int main()
{
    int num, shifts;

    printf("Enter the number: ");

    scanf("%d", &num);

    while (num > 0)
    {
        num <<= 1;

        shifts++;
    }

    printf("number of shift is %d", shifts);

    return 0;
}
```

8) Write a C program that creates a bitmask with the first n bits set to 1 using the left shift operator.

```
#include <stdio.h>

int main()
{
    int n, res;

    printf("Enter value of n:");

    scanf("%d", &n);

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

    printf("The bitmask is %d", res);

}
```

9) Develop a C program that reverses the bits of an integer using left shift and right shift operations.

```
#include <stdio.h>

int main()
{
    unsigned int num, rev = 0;
    printf("Enter the number: \n");
    scanf("%u",&num);
    for (int i = 0; i < 32; i++) {
        rev = rev << 1;
        rev = rev|(num & 1);
        num = num >> 1;
    }
    printf("Reversed bits: %u\n", rev);
    return 0;
}
```

10) Create a C program that performs a circular left shift on an integer.

```
#include <stdio.h>

int main()
{
    unsigned int num,res;
    int shift;
    printf("Enter a number: ");
    scanf("%u", &num);
    printf("Enter shift amount: ");
    scanf("%d", &shift);
```

```

    res = (num << shift) | (num >> (32 - shift));

    printf("after shift: %u", res);

    return 0;
}

```

11) Write a C program that takes an integer input and divides it by  $2^n$  using the right shift operator.

```

#include<stdio.h>

int main()
{
    int num,n,res;

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

    scanf("%d",&num);

    printf("Enter n: ");

    scanf("%d",&n);

    res=num>>n;

    printf("%d",res);

    return 0;
}

```

12) Create a C program that counts how many times you can right shift a number before it becomes zero.

```

#include <stdio.h>

int main()
{
    int num,shifts;

    printf("Enter the number: ");

    scanf("%d", &num);

    while (num > 0)
    {

```

```

    num >>= 1;

    shifts++;

}

printf("number of shift is %d", shifts);

return 0;

}

```

13) Write a C program that extracts the last n bits from a given integer using the right shift operator.

```

#include<stdio.h>

int main()
{
    int num,n,res,mask;

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

    scanf("%d",&num);

    printf("Enter n: ");

    scanf("%d",&n);

    mask=(1<<n)-1;

    res==num & mask;

    printf("%d",res);

    return 0;

}

```

14) Develop a C program that uses the right shift operator to create a bitmask that checks if specific bits are set in an integer.

```

#include <stdio.h>

int main() {

    int num, n,mask;

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

```

```
scanf("%d",&num);  
printf("Enter n: ");  
scanf("%d",&n);  
mask = num >> n;  
if (mask & 1)  
{  
    printf("%d bit is set\n", n);  
}  
else  
{  
    printf("%d bit is not set\n", n);  
}  
return 0;  
}
```