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 to check even or odd\n");
   scanf("%d",&num);
   if((num&1)){
      printf("%d is odd\n",num);
   }else{
      printf("%d is even\n",num);
   }
}
```

Output

```
enter the number to check even or odd
5
5 is odd
enter the number to check even or odd
8
8 is even
```

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

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

   printf("enter the number\n");
   scanf("%d",&num);
   printf("enter the index to check\n");
   scanf("%d",&n);
   int result=((num>>n)&1);
   printf("the value of %d th bit is %d\n",n,result);
   return 0
  }
```

Output

```
enter the number
10
enter the index to check
1
the value of 1 th bit is 1
```

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

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

   printf("enter the number\n");
   scanf("%d",&num);
   printf("enter the index to set the value to 1\n");
   scanf("%d",&n);
   int result=(num|(1<<n));    //(1<<n) is to find mask value
   printf("The result after setting the bit position %d to 1 is:%d",n,result);
   return 0;
}
```

Output

```
enter the number
8
enter the index to set the value to 1
1
The result after setting the bit position 1 to 1 is:10
```

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;
     printf("enter the number\n");
    scanf("%d",&num);
    printf("enter the index to clear the value \n");
    scanf("%d",&n);

    int result=(num&(~(1<<n)));
    printf("The result after setting the bit position %d to 0 is:%d",n,result);

}
```

Output

```
enter the number
15
enter the index to clear the value
3
The result after setting the bit position 3 to 0 is:7
```

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

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

     printf("enter the number\n");
    scanf("%d",&num);

    printf("enter the index to toggle\n");
    scanf("%d",&n);
    int result=(num^(1<<n));
    printf("result=%d",result);
    return 0;
}
```

```
Output
enter the number
10
enter the index to toggle
1
result=8
```

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, result;
    printf("Enter an integer: ");
    scanf("%d", &num);
    printf("Enter the value of n: ");
    scanf("%d", &n);

    result = num << n;



    printf("RESULT=%d",result);


    return 0;
}
```

```
Output

Enter an integer: 3

Enter the value of n: 2

RESULT=12
```

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;

   int count=0;

   printf("enter a number to check");

   scanf("%d",&num);

   while(num>0){

      num<<=1;

      count=count+1;

         }

   printf("limit=%d",count);

   }
```

```
Output

enter a number to check:

4

limit=29
```

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 num;

   printf("enter a number to check:\n");

   scanf("%d",&num);

   int bitmask=(1<<num)-1;
```

```
    printf("Bitmask=%d",bitmask);
    return 0;
    }
```

Output

```
enter a number to check:
5
Bitmask=31
```

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

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

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

```
#include<stdio.h>
#include<limits.h>
int main(){
    int num,n,result;
    printf("enter a number\n");
    scanf("%d",&num);
    printf("enter the value of n\n");
    scanf("%d",&n);
    result=num>>n;
    printf("result=%d",result);
}
```

Output

```
enter a number
16
enter the value of n
2
result=4
```

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;

   int count=0;

   printf("enter a number to check:\n");

   scanf("%d",&num);

   while(num>0){

       num>>=1;

       count=count+1;


   }

   printf("limit=%d",count);


}
```

Output

```
enter a number to check:

18

limit=5
```

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;

    printf("enter a number:\n");

    scanf("%d",&num);

    printf("enter n:\n");

    scanf("%d",&n);

     int result=num&((1<<n)-1);

    printf("Last %d bits are: %d",n,result);  }
```

Output

```
enter a number:

29

enter n:

3

Last 3 bits are: 5
```

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.

```c
#include <stdio.h>

int main(){

   int num,n;

   printf("enter a number:\n");

   scanf("%d",&num);

   printf("enter bit position to check:\n");

   scanf("%d",&n);


   if ((num>>1)&1){

      printf("Bit %d is set to 1",n);

   }else{

      printf("bit %d is not set 0",n);

   }

  return 0;


}
```

Output

```
enter a number:

18

enter bit position to check:

1

Bit 1 is set to 1
```