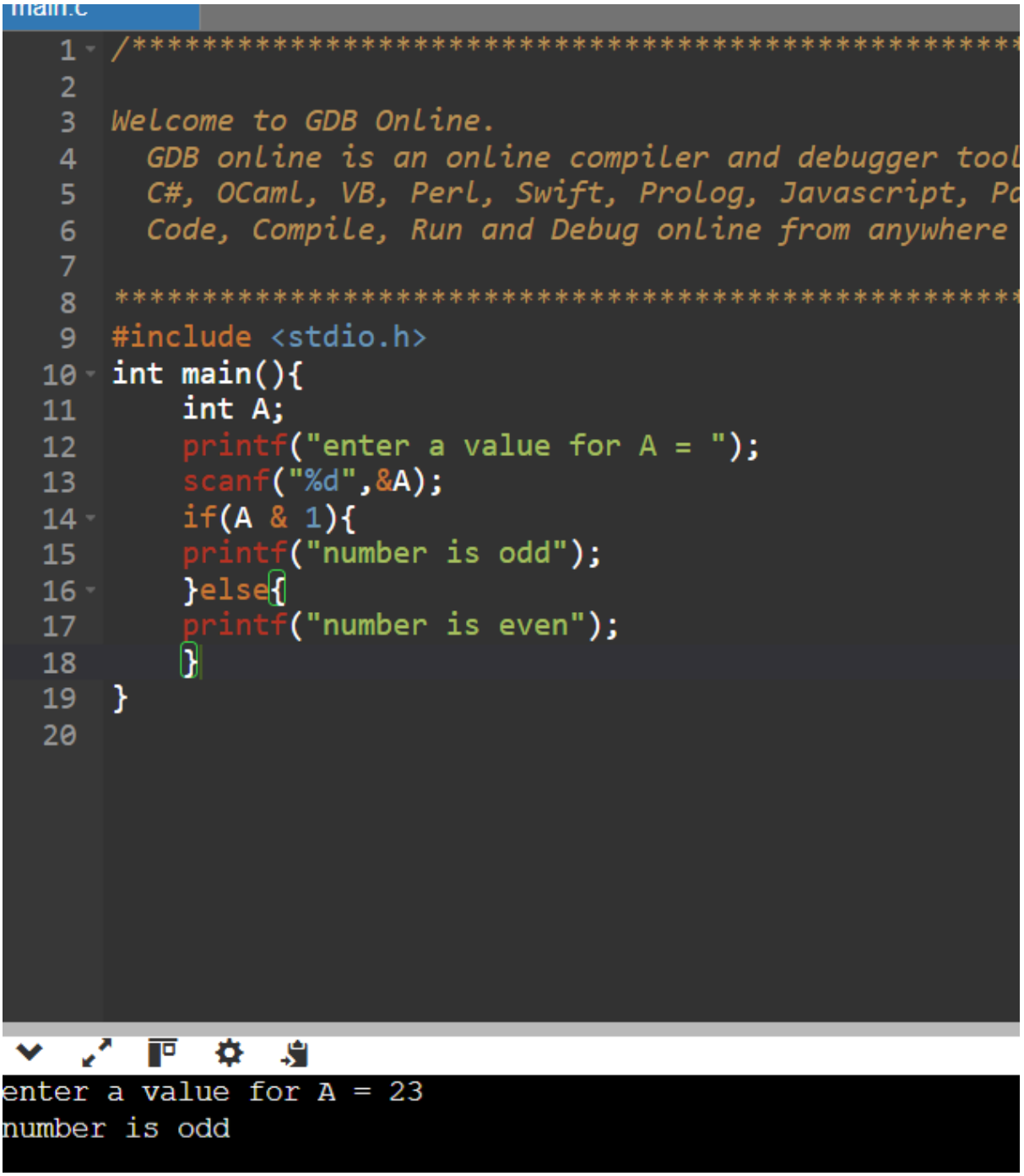
1. Write a C program to determine if the least significant bit of a given integer is set (i.e., check if the number is odd).



```
/*******************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pa
  Code, Compile, Run and Debug online from anywhere

*******************************************************/
#include <stdio.h>
int main(){
    int A;
    printf("enter a value for A = ");
    scanf("%d",&A);
    if(A & 1){
    printf("number is odd");
    }else{
    printf("number is even");
    }
}

enter a value for A = 23
number is odd
```

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

main.c

```c
/*****************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pa
  Code, Compile, Run and Debug online from anywhere

*****************************************************/
#include <stdio.h>
int main(){
    int A,n;
    printf("enter a value for A = ");
    scanf("%d",&A);
    printf("enter the nth position ");
    scanf("%d",&n);

    if (A & (1 << n)) {
        printf("The %d th bit is 1\n", n);
    } else {
        printf("The %d th bit is 0\n", n);
    }

    return 0;
}
```

```
enter a value for A = 45
enter the nth position 6
The 6 th bit is 0
```

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

main.c

```c
/*****************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool fo
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pasca
  Code, Compile, Run and Debug online from anywhere in

*****************************************************/
#include <stdio.h>
int main(){
    int A,n;
    printf("enter a value for A = ");
    scanf("%d",&A);
    printf("enter the nth position ");
    scanf("%d",&n);

    A = A | (1 << n);
    printf("The %d th bit is updated as 1\n", n);
    printf("updated value = %d\n",A);

    return 0;
}

```

```
enter a value for A = 34
enter the nth position 4
The 4 th bit is updated as 1
updated value = 50
```

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

main.c

```c
/*****************************************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool for C, C++, Python, PHP, R
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
  Code, Compile, Run and Debug online from anywhere in world.

*****************************************************************************/
#include <stdio.h>
int main(){
    int A,n;
    printf("enter a value for A = ");
    scanf("%d",&A);
    printf("enter the nth position ");
    scanf("%d",&n);

    int mask = ~(1<<n);
    A = A & mask;
    printf("The %d th bit is cleared and updated value of A = %d\n", n,A);

    return 0;
}
```

input

```
enter a value for A = 54
enter the nth position 4
The 4 th bit is cleared and updated value of A = 38
```

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

main.c

```c
/******************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger too
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, P
  Code, Compile, Run and Debug online from anywhere

******************************************************/
#include <stdio.h>
int main(){
    int A,n;
    printf("enter a value for A = ");
    scanf("%d",&A);
    printf("enter the nth position ");
    scanf("%d",&n);

    A = A ^ (1 << n);
    printf("After toggling A = %d\n",A);

    return 0;
}

```

```
enter a value for A = 55
enter the nth position 5
After toggling A = 23
```

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

main.c

```c
/*****************************************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool for C, C++, Python, PHP, R
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
  Code, Compile, Run and Debug online from anywhere in world.

*****************************************************************************/
#include <stdio.h>

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

    printf("Enter an integer: ");
    scanf("%d", &num);

    printf("enter value of n : ");
    scanf("%d", &n);

   out = num << n;

    printf("The result of %d * 2^%d is: %d\n", num, n, out);

    return 0;
}
```

input

```
Enter an integer: 267
enter value of n : 2
The result of 267 * 2^2 is: 1068
```

2. 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).

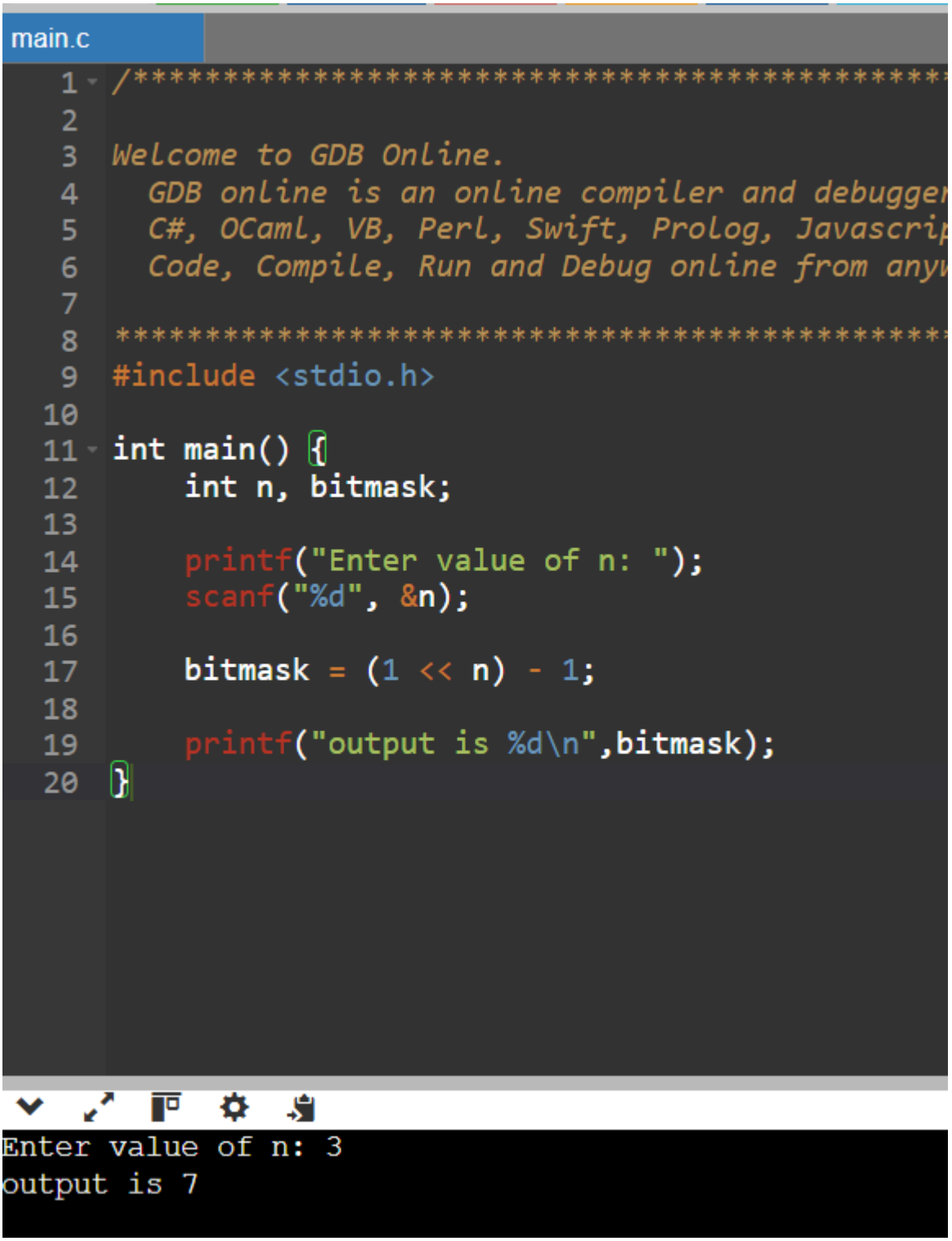
main.c

```c
/*****************************************************************************/
#include <stdio.h>
#include <limits.h>

#include <stdio.h>

int count(int num) {
    int count = 0;
    while (num != 0) {
        num <<= 1;
        count++;}
   return count;
}

int main() {
    int num;

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

    int shifts = count(num);

    printf("The number can be left shifted %d times before it overflows.\n", shifts);

    return 0;
}
```

input

```
Enter a number: 45
The number can be left shifted 32 times before it overflows.
```

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



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

main.c

```c
#include <stdio.h>

 int reverse_bits( int num) {
 int reverse = 0;
    int count = sizeof(num) * 8;
    for (int i = 0; i < count; i++) {
        reverse <<= 1;
        reverse |= (num & 1);
        num >>= 1;
    }

    return reverse;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%u", &num);
    int reverse = reverse_bits(num);
    printf("Original number: %u\n", num);
    printf("Reversed number: %u\n", reverse);

    return 0;
}
```

```
Enter a number: 45
Original number: 45
Reversed number: 3019898880
```

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

main.c

```c
/*******************************************************************

Welcome to GDB Online.
  GDB online is an online compiler and debugger tool for C, C++, P
  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, H
  Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************/
#include <stdio.h>

int main() {
    int num, n , out;
    printf("Enter an integer: ");
    scanf("%d", &num);
    printf("Enter n ");
    scanf("%d", &n);

    out = num >> n;

    printf("Result of %d divided by 2^%d = %d\n", num, n, out);
    return 0;
}
```

input

```
Enter an integer: 20
Enter n 3
Result of 20 divided by 2^3 = 2
```

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

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

main.c

```
  Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************/
#include <stdio.h>
#include <stdint.h>

uint16_t lastbits(uint16_t num, int n) {
    uint16_t mask = (1 << n) - 1;

    return num & mask;
}

int main() {
    uint16_t num;
    int n;

    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Enter the number of bits to extract: ");
    scanf("%d", &n);

    uint16_t last_n_bits = lastbits(num, n);

    printf("Last %d bits of %hu are: %hu\n", n, num, last_n_bits);

    return 0;
}
```

input

```
      |             %hd
Enter a number: 23
Enter the number of bits to extract: 3
Last 3 bits of 23 are: 7
```