

Usage of static

```
#include <stdio.h>
void myFun(void);
int main()
{
    myFun();
    myFun();
    myFun();
    myFun();
    //printf("002 The function is ececuted  %d times\n",count);
    return 0;
}
void myFun()
{
    static int count = 0;
    count = count + 1;
    printf("001 The function is ececuted  %d times\n",count);
}
```

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 a number\n");
    scanf("%d",&num);
    if(num&1)
    {
        printf("The LSB is set as the num is odd");
    }
    else
    {
        printf("The LSB is not set as the num is even");
    }
    return 0;
}
```

input

```
enter a number
13
The LSB is set as the num is odd
```

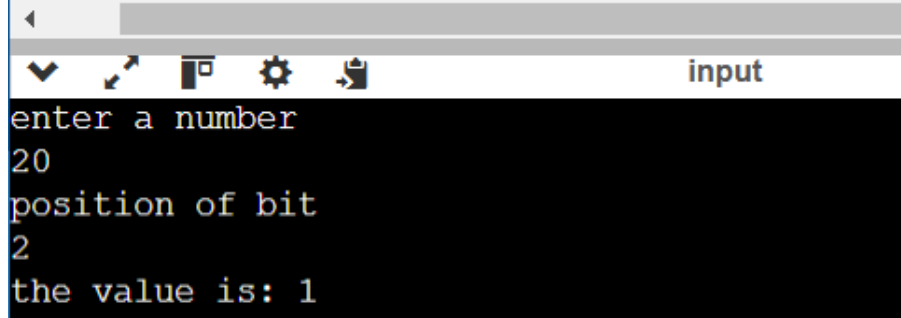
```
enter a number
26
The LSB is not set as the num is even

...Program finished with exit code 0
Press ENTER to exit console.
```

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

```
#include <stdio.h>

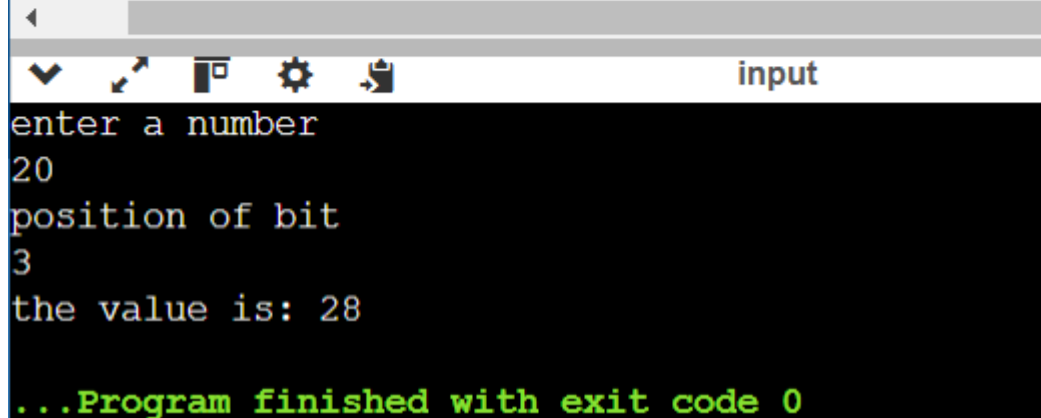
int main()
{
    int num,n;
    printf("enter a number\n");
    scanf("%d",&num);
    printf("position of bit\n");
    scanf("%d",&n);
    int bitvalue = (num >> n) & 1;
    printf("the value is: %d",bitvalue);
    return 0;
}
```

The image shows a terminal window with a dark background. At the top, there is a toolbar with icons for a dropdown menu, a cursor, a window, a gear, and a clipboard. To the right of the toolbar, the word "input" is displayed. Below the toolbar, the program's output is shown in a monospaced font. The prompts and user input are as follows:
enter a number
20
position of bit
2
the value is: 1

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

```
#include <stdio.h>

int main()
{
    int num,n;
    printf("enter a number\n");
    scanf("%d",&num);
    printf("position of bit\n");
    scanf("%d",&n);
    num = num|(1<<n);
    printf("the value is: %d",num);
    return 0;
}
```



The screenshot shows a terminal window with a toolbar at the top containing icons for a dropdown menu, a cursor, a window, a gear, and a clipboard. The text in the terminal is as follows:

```
enter a number
20
position of bit
3
the value is: 28
```

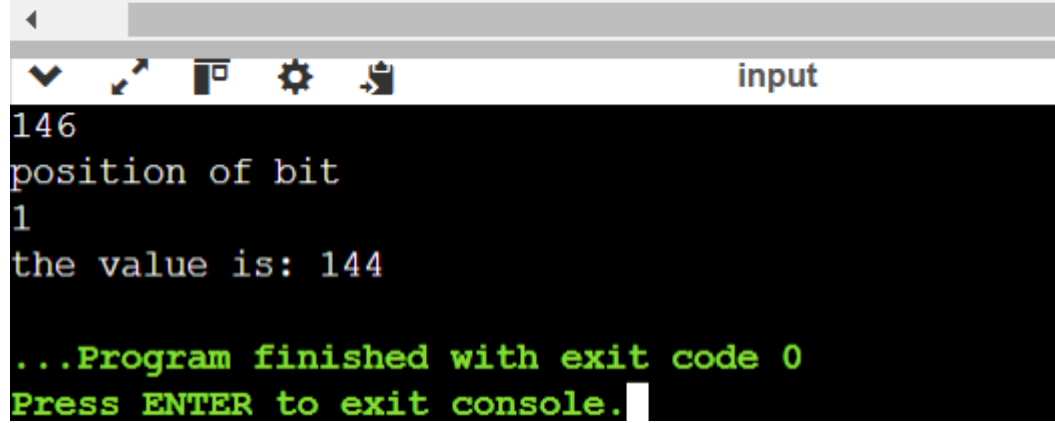
...Program finished with exit code 0

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;
    printf("enter a number\n");
    scanf("%d",&num);
    printf("position of bit\n");
    scanf("%d",&n);

    num = num & (~ (1 << n));
    printf("the value is: %d",num);
    return 0;
}
```



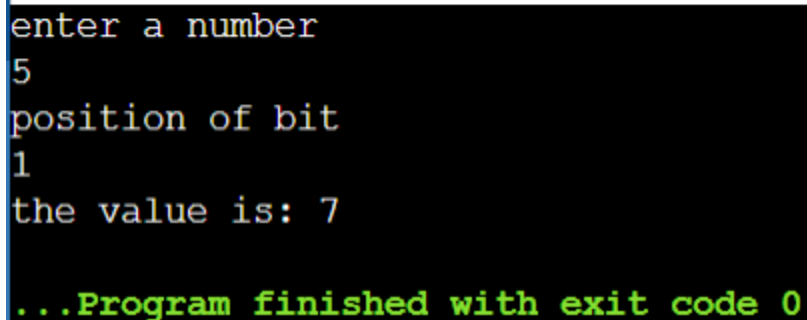
146
position of bit
1
the value is: 144
...Program finished with exit code 0
Press ENTER to exit console.

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

```
#include <stdio.h>

int main()
{
    int num,n;
    printf("enter a number\n");
    scanf("%d",&num);
    printf("position of bit\n");
    scanf("%d",&n);

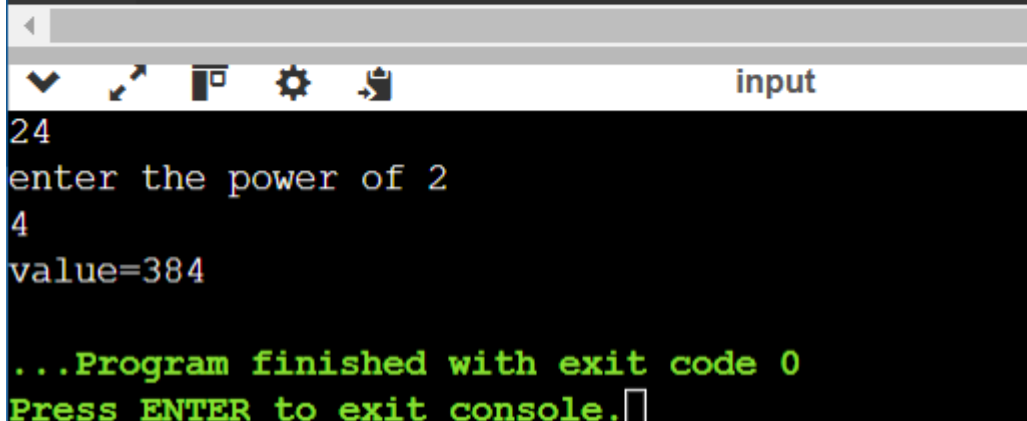
    num = num^(1<<n);
    printf("the value is: %d",num);
    return 0;
}
```



enter a number
5
position of bit
1
the value is: 7
...Program finished with exit code 0

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

```
9  #include <stdio.h>
10
11 int main()
12 {
13     int x,n;
14     printf("enter the integer\n");
15     scanf("%d",&x);
16     printf("enter the power of 2\n");
17     scanf("%d",&n);
18
19     int res=x<<n;
20     printf("value=%d",res);
21 }
```

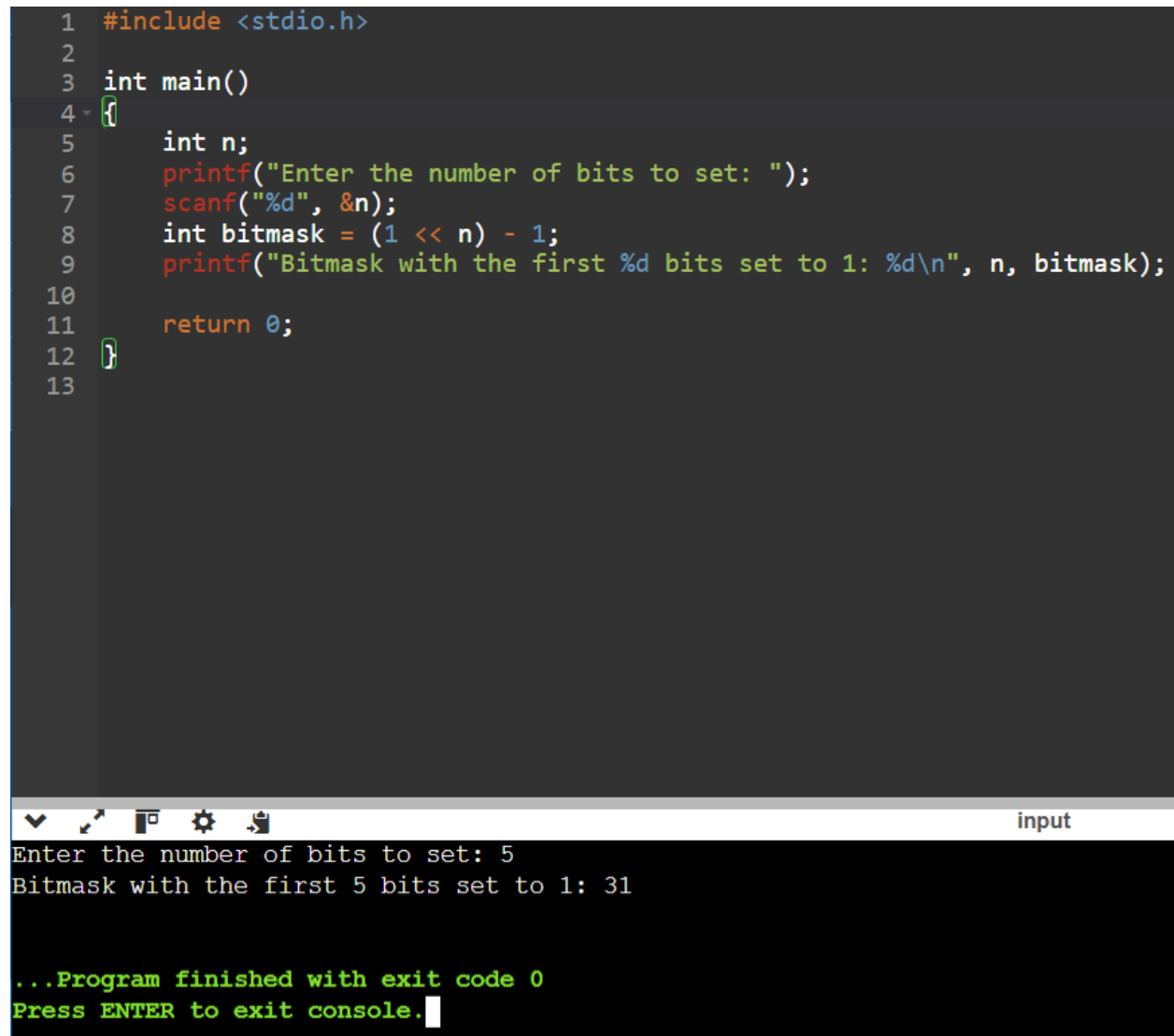


24
enter the power of 2
4
value=384
...Program finished with exit code 0
Press ENTER to exit console.

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).

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

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      printf("Enter the number of bits to set: ");
7      scanf("%d", &n);
8      int bitmask = (1 << n) - 1;
9      printf("Bitmask with the first %d bits set to 1: %d\n", n, bitmask);
10
11     return 0;
12 }
13
```

The image shows a code editor window with a dark background and a terminal window below it. The code editor contains a C program that prompts the user for the number of bits to set, calculates a bitmask using the left shift operator, and prints the result. The terminal window shows the program's execution with the input '5' and the output '31'. The terminal also shows the program finished with exit code 0 and a prompt to press ENTER to exit the console.

input

Enter the number of bits to set: 5
Bitmask with the first 5 bits set to 1: 31

...Program finished with exit code 0
Press ENTER to exit console.

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.

```
1  #include <stdio.h>
2  int main()
3  {
4      unsigned int n, shift;
5      printf("Enter a number: ");
6      scanf("%u", &n);
7      printf("Enter shifts: ");
8      scanf("%u", &shift);
9      unsigned int res = (n << shift) | (n >> (32 - shift));
10     printf("Result: %u\n", res);
11     return 0;
12 }
```

Enter a number: 7

Enter shifts: 3

Result: 56

...Program finished with exit code 0

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

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x, n;
6      printf("Enter the integer: ");
7      scanf("%d", &x);
8      printf("Enter the power of 2: ");
9      scanf("%d", &n);
10     int result = x >> n;
11     printf("Result = %d", result);
12
13     return 0;
14 }
15
```

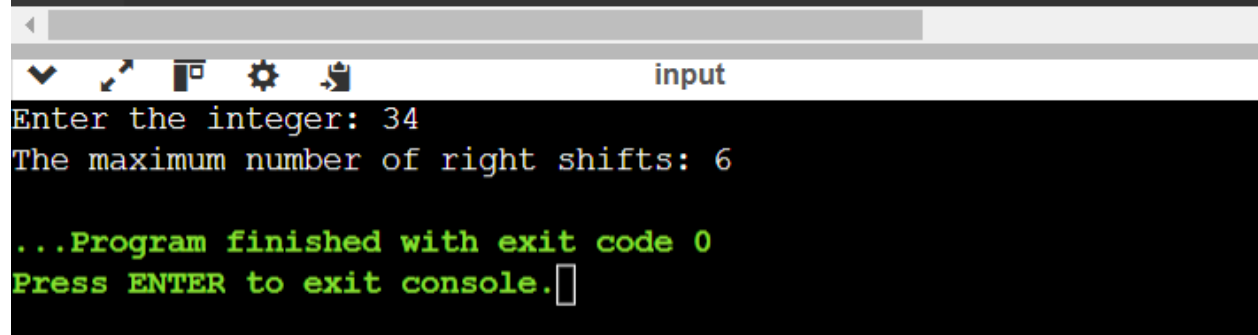
input

```
Enter the integer: 24
Enter the power of 2: 2
Result = 6
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

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

```
1  #include <stdio.h>
2
3  int main() {
4      int c=0;
5      int n;
6      printf("Enter the integer: ");
7      scanf("%d", &n);
8      while (n != 0) {
9          n = n >> 1;
10         c++;
11     }
12     printf("The maximum number of right shifts: %d", c);
13
14     return 0;
15 }
16
```



The screenshot shows a terminal window with a toolbar at the top containing icons for a dropdown menu, zoom in, zoom out, settings, and a file icon. The window title is "input". The terminal output shows the program's execution: it prompts for an integer, receives 34, and outputs that the maximum number of right shifts is 6. It then displays the exit code and a prompt to press ENTER to exit the console.

```
input
Enter the integer: 34
The maximum number of right shifts: 6

...Program finished with exit code 0
Press ENTER to exit console.
```

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

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x, n;
6      printf("Enter the integer: ");
7      scanf("%d", &x);
8      printf("Enter the number of bits: ");
9      scanf("%d", &n);
10     int result = x & ((1 << n) - 1);
11     printf("Last %d bits of %d are: %d\n", n, x, result);
12
13     return 0;
14 }
15
```

input

```
Enter the integer: 30
Enter the number of bits: 4
Last 4 bits of 30 are: 14

...Program finished with exit code 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.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x, pos;
6      printf("Enter the integer: ");
7      scanf("%d", &x);
8      printf("Enter the bit position to check: ");
9      scanf("%d", &pos);
10
11     int mask = x >> pos;
12     if (mask & 1)
13     {
14         printf("Bit at position %d is set\n", pos);
15     }
16     else
17     {
18         printf("Bit at position %d is not set\n", pos);
19     }
20
21     return 0;
22 }
23
```

input

Enter the integer: 36
Enter the bit position to check: 5
Bit at position 5 is set

Enter the integer: 36
Enter the bit position to check: 4
Bit at position 4 is not set

...Program finished with exit code 0