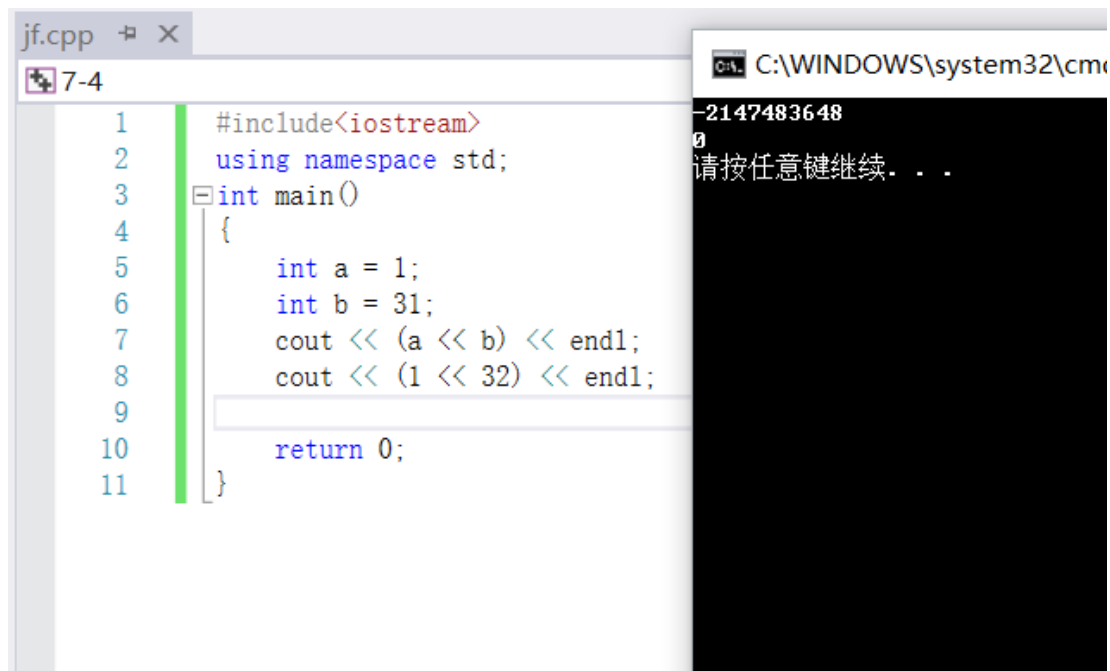


<朱世轩 计 2 1752528>

1. warning 的意思：移位操作的位数超过了 int 型数据的位数，是未定义的操作
2. 得到该结果的原因：编译器将超过 31 的 b 进行了对 32（整形位数）取余的处理（与 31 进行&运算），得到 0 后再进行移位，所以  $(a \ll b)$  的结果仍是 1；而 1 和 32 是常量，直接进行移位操作，所以得到了左移 32 位后的结果 0。

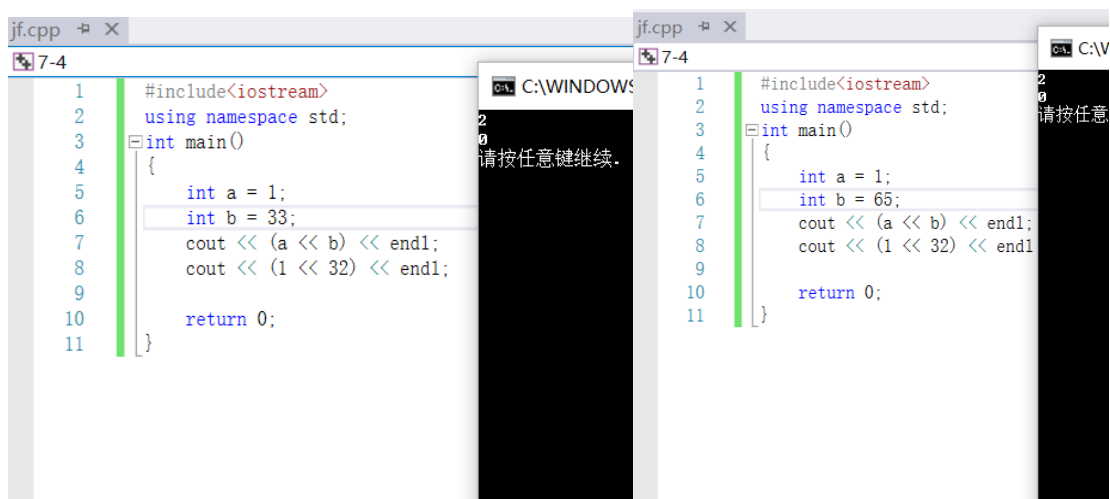


The screenshot shows a C++ IDE with a file named jf.cpp. The code is as follows:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a = 1;
6     int b = 31;
7     cout << (a << b) << endl;
8     cout << (1 << 32) << endl;
9
10    return 0;
11 }
```

The output window shows the following results:

```
-2147483648
0
请按任意键继续. . .
```



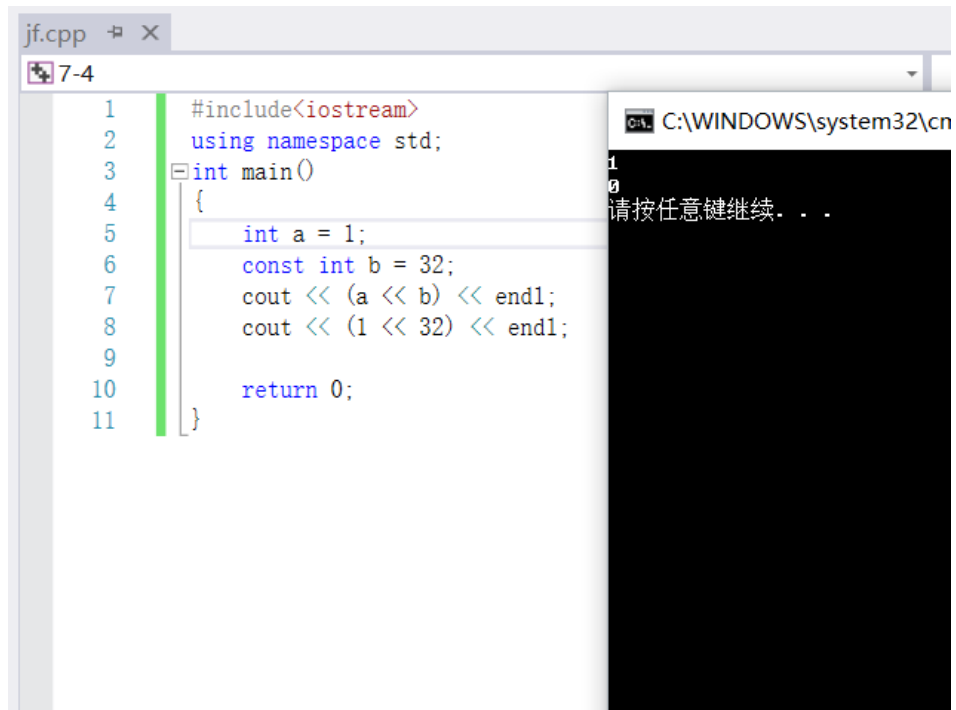
The top screenshot shows the same code as above but with `b = 33`. The output window shows:

```
2
0
请按任意键继续.
```

The bottom screenshot shows the same code with `b = 65`. The output window shows:

```
2
0
请按任意键继续.
```

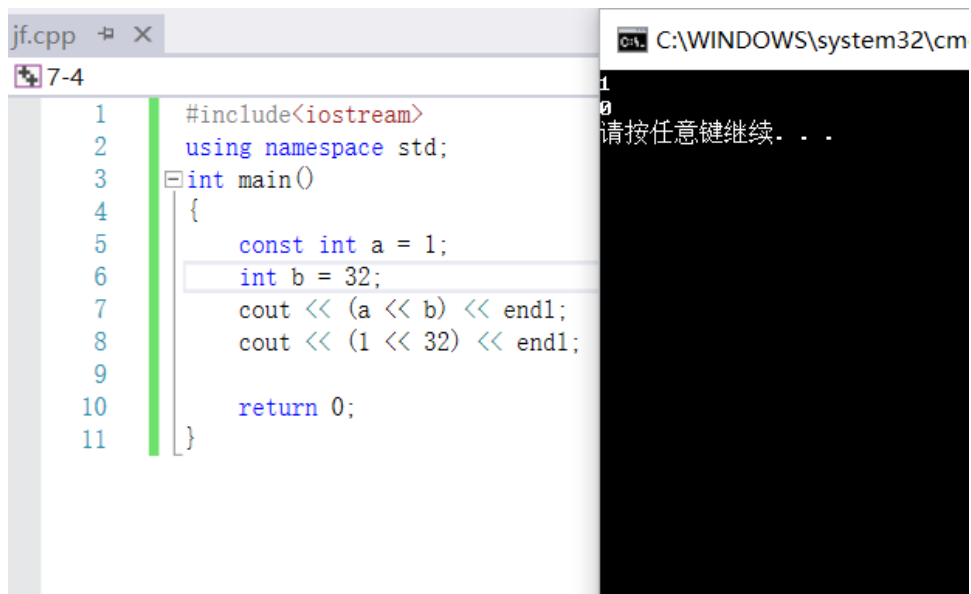
分别将 b 的值改为 31, 33, 65, 发现在 32 以内的移位是正常的, 超过 32 的移位是对 b 进行了对 32 取余的处理



```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 1;
6      const int b = 32;
7      cout << (a << b) << endl;
8      cout << (1 << 32) << endl;
9
10     return 0;
11 }
```

C:\WINDOWS\system32\cmd.exe

```
1
0
请按任意键继续. . .
```



```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      const int a = 1;
6      int b = 32;
7      cout << (a << b) << endl;
8      cout << (1 << 32) << endl;
9
10     return 0;
11 }
```

C:\WINDOWS\system32\cmd.exe

```
1
0
请按任意键继续. . .
```

可以看到，将 a，b 其中一个改为常变量后，编译器仍会执行同样的将 b 对 32 取余操作

The screenshot shows a C++ IDE with two windows. The left window, titled 'jf.cpp', contains the following code:

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      const int a = 1;
6      const int b = 32;
7      cout << (a << b) << endl;
8      cout << (1 << 32) << endl;
9
10     return 0;
11 }
```

The right window, titled 'C:\WINDOWS\system32\cmd.exe', shows the output of the program:

```
0
0
请按任意键继续. . .
```

而将 a, b 都改为常变量后, 与直接将 1 移位 32 位效果相同

在 codeblocks 和 dev-c++ 中的测试结果与在 VS 中相同