# 位操作

[位或&](#_位或&) [位与|](#_位与|) [位非~](#_位非~) [左移<<](#_左移<<) [右移>>](#_右移>>)

[通过位操作实现操作示例](#_通过位操作实现操作示例)

# 位或&

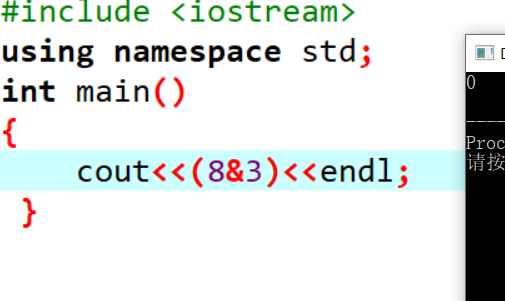
|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

例子:  
 8&3=0

8: 0000 1000

3: & 0000 0011

0000 0000



# 位与|

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

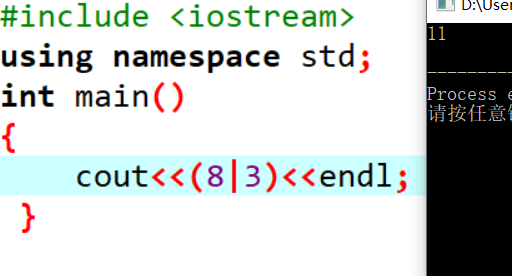
例子:

8|3=11

8: 0000 1000

3: | 0000 0011

0000 1011



# 位非~

|  |  |
| --- | --- |
| 1 | 0 |
| 0 | 1 |

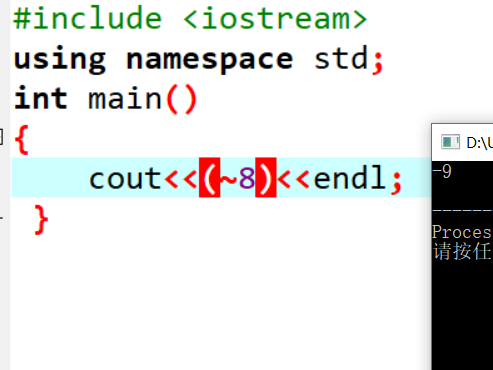
例子:

~8=-9

8: 0000 1000

|

1111 0111



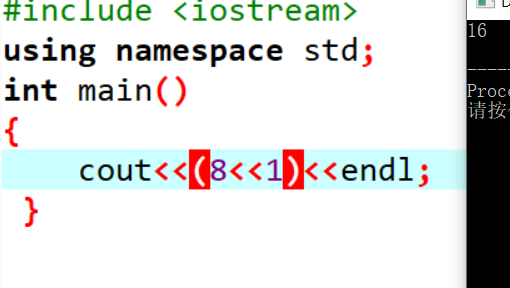
# 左移<<

8<<1=16

8: 0000 1000

<< 00001 0000 (黄色溢出,绿色新增)

0001 0000



# 右移>>

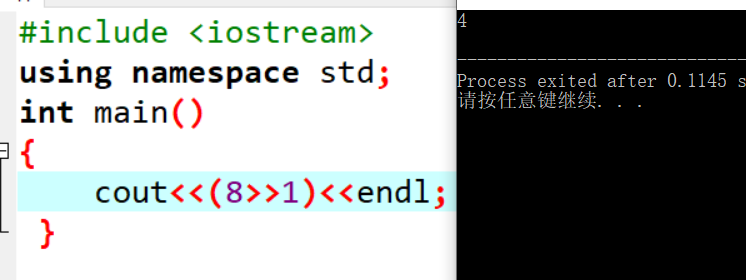
例子:

8>>1=4

8: 0000 1000

>> 0000 01000 (黄色溢出,绿色新增)

0000 0100



# 通过位操作实现操作示例

需求分析:由用户输入一个数,把这个数的最低4位变成0100

程序实现思路分析:把最低4位先清0(位或),再置数(位与)

示例:用户输入8

0000 1000

& 1111 0000

0000 0000

| 0000 0110

0000 0110

代码演示:

#include <iostream>

using namespace std;

int main()

{

cout<<"请输入一个整数";

int a;

cin>>a;

a=a&~15;//此处用~15代替1111 0000

a=a|6;

cout<<a;

}