Weifan Lin

Csc342 Section G

02/23/2015

Homework

Title:

Overflow In Programing

Objective:

The goal is to understand when would an overflow occur in programing.

Design in C++:

Declare a variable and ask user to input the 32-bit largest positive number.

Print the representations of Binary, Hexadecimal, and decimal of the input.

Add 1 to the input and see what happens.

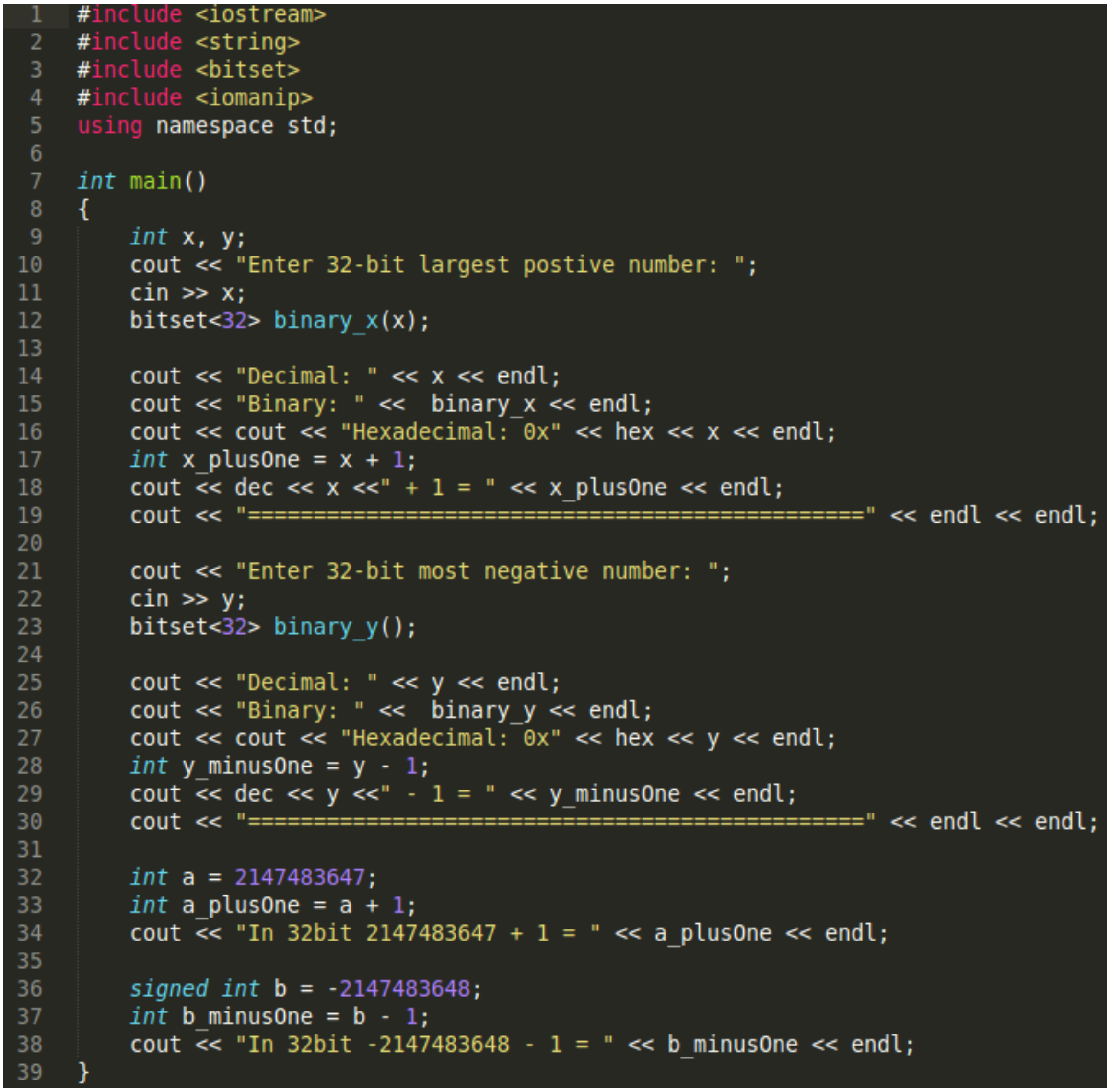
Declare another variable and ask user to input the 32-bit most negative number.

Print the representations of Binary, Hexadecimal, and decimal of the input.

Minus 1 from the input and see what happens.

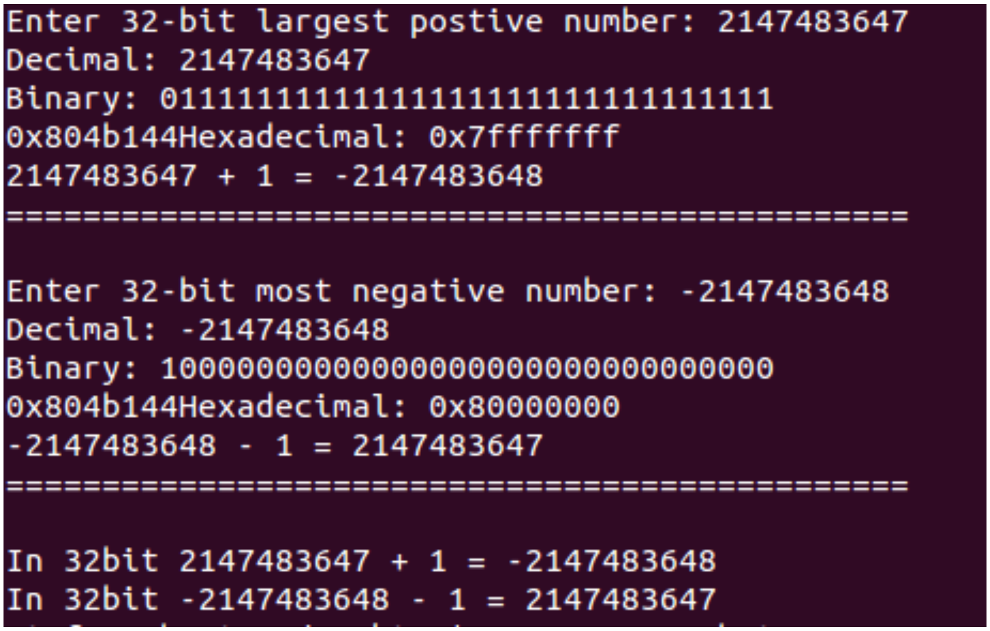
Assign a largest positive number to a variable, add 1 to it and see what happens.

Assign a most negative number to another variable, minus 1 from it and see what happens.



```cpp
#include <iostream>
#include <string>
#include <bitset>
#include <iomanip>
using namespace std;

int main()
{
    int x, y;
    cout << "Enter 32-bit largest postive number: ";
    cin >> x;
    bitset<32> binary_x(x);

    cout << "Decimal: " << x << endl;
    cout << "Binary: " <<  binary_x << endl;
    cout << cout << "Hexadecimal: 0x" << hex << x << endl;
    int x_plusOne = x + 1;
    cout << dec << x <<" + 1 = " << x_plusOne << endl;
    cout << "===============================================" << endl << endl;

    cout << "Enter 32-bit most negative number: ";
    cin >> y;
    bitset<32> binary_y();

    cout << "Decimal: " << y << endl;
    cout << "Binary: " <<  binary_y << endl;
    cout << cout << "Hexadecimal: 0x" << hex << y << endl;
    int y_minusOne = y - 1;
    cout << dec << y <<" - 1 = " << y_minusOne << endl;
    cout << "===============================================" << endl << endl;

    int a = 2147483647;
    int a_plusOne = a + 1;
    cout << "In 32bit 2147483647 + 1 = " << a_plusOne << endl;

    signed int b = -2147483648;
    int b_minusOne = b - 1;
    cout << "In 32bit -2147483648 - 1 = " << b_minusOne << endl;
}
```

Now we compile this C++ file using 32-bit g++ compiler:



```
Enter 32-bit largest postive number: 2147483647
Decimal: 2147483647
Binary: 01111111111111111111111111111111
0x804b144Hexadecimal: 0x7fffffff
2147483647 + 1 = -2147483648
===============================================

Enter 32-bit most negative number: -2147483648
Decimal: -2147483648
Binary: 10000000000000000000000000000000
0x804b144Hexadecimal: 0x80000000
-2147483648 - 1 = 2147483647
===============================================

In 32bit 2147483647 + 1 = -2147483648
In 32bit -2147483648 - 1 = 2147483647
```

As we can see clearly from the result, in 32-bit world if we add 1 to the largest positive number 2147483647 we would have -2147483648 as result which is an overflow, and if we minus 1 from the most negative number -2147483648 we would have 2147483647 as result which also is an overflow.