

# BigInteger

```

#include <string>
#include <sstream>
#include <map>
#include <vector>
#include <iostream>
#include <stdio.h>
#include <string.h>

using namespace std;

struct BigInteger {
    static const int BASE = 100000000;
    static const int WIDTH = 8;
    vector<int> s;

    BigInteger(long long num = 0) { *this = num; }
    BigInteger operator = (long long num) {
        s.clear();
        do {
            s.push_back(num % BASE);
            num /= BASE;
        } while(num > 0);
        return *this;
    }
    BigInteger operator = (const string& str) {
        s.clear();
        int x, len = (str.length() - 1) / WIDTH + 1;
        for(int i = 0; i < len; i++) {
            int end = str.length() - i*WIDTH;
            int start = max(0, end - WIDTH);
            sscanf(str.substr(start, end-start).c_str(), "%d", &x);
            s.push_back(x);
        }
        return *this;
    }
    BigInteger operator + (const BigInteger& b) const {
        BigInteger c;
        c.s.clear();
        for(int i = 0, g = 0; ; i++) {
            if(g == 0 && i >= s.size() && i >= b.s.size()) break;
            int x = g;
            if(i < s.size()) x += s[i];
            if(i < b.s.size()) x += b.s[i];
            c.s.push_back(x % BASE);
            g = x / BASE;
        }
        return c;
    }
};

ostream& operator << (ostream &out, const BigInteger& x) {
    out << x.s.back();
    for(int i = x.s.size()-2; i >= 0; i--) {

```

```
    char buf[20];
    sprintf(buf, "%08d", x.s[i]);
    for(int j = 0; j < strlen(buf); j++) out << buf[j];
}
return out;
}

istream& operator >> (istream &in, BigInteger& x) {
    string s;
    if(!(in >> s)) return in;
    x = s;
    return in;
}

int main(void){
    BigInteger a, b, c;
    cin >> a >> b;
    c = a + b;
    cout << a << "\n";
    cout << b << "\n";
    cout << c << "\n";
    return 0;
}
```