9523 MyString
9531 看上去好坑的运算符重载
9533 惊呆！Point 竟然能这样输入输出
8002 第四周程序填空题 3(二维数组类 Array2)
9534 别叫，这个大整数已经很简化了！

****8002.cpp:

```
#include <iostream>
using namespace std;

/*
用一维数组来存放二维数组

a[i][j]的计算过程从左到右，
a[i] 的返回值是个指针，
指向第 i 行的首地址。
a[i][j] 就会是第 i 行第 j 个元素了。
*/



class Array2
{
//your code starts here
private:
     int * p;
     int r,c;
public:
    Array2() { p = NULL ; }
    Array2( int r_, int c_  ):r(r_),c(c_)
    {     p = new int [ r * c];    }
    Array2( Array2 & a ):r(a.r),c(a.c)
    {
        p = new int [r * c];
        memcpy( p, a.p, sizeof(int )*r*c);
    }
    Array2 & operator=(const Array2 & a) {
        if( p )
            delete [] p;
        r = a.r; c = a.c;p = new int [r * c];
        memcpy( p, a.p, sizeof(int ) * r * c);
        return * this;
   }
     ~Array2()
    {
        if( p)
            delete [] p;
    }
    int * operator [] ( int i )     {
        return p + i * c;
    }
    int & operator() ( int i,int j )     {
        return p[ i * c + j];
    }
//your code ends here
};
int main() {
    Array2 a(3,4);
    int i,j;
    for(  i = 0;i < 3; ++i )
        for(  j = 0; j < 4; j ++ )
            a[i][j] = i * 4 + j;
    for(  i = 0;i < 3; ++i ) {
        for(  j = 0; j < 4; j ++ ) {
            cout << a(i,j) << ",";
        }
        cout << endl;
    }
    cout << "next" << endl;
    Array2 b;     b = a;
    for(  i = 0;i < 3; ++i ) {
        for(  j = 0; j < 4; j ++ ) {
            cout << b[i][j] << ",";
        }
        cout << endl;
    }
    return 0;
}
```

****9523.cpp:

/*

补足MyString类，使程序输出指定结果

输入

多组数据，每组一行，是两个不带空格的字符串

输出

对每组数据，先输出一行，打印输入中的第一个字符串三次
然后再输出一行，打印输入中的第二个字符串三次

输入样例

```
abc def
123 456
```

输出样例

```
abc,abc,abc
def,def,def
123,123,123
456,456,456
```

```
*/
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
class MyString {
    char * p;
public:
    MyString(const char * s) {
        if( s) {
            p = new char[strlen(s) + 1];
            strcpy(p,s);
        }
        else
            p = NULL;

    }
    ~MyString() { if(p) delete [] p; }
//your code starts here

    friend ostream & operator << (ostream & o,const MyString & s) {
        o << s.p;
        return o;
    }
    void Copy(const char * s) {
        if( p )
            delete [] p;

        if( s ) {
            p = new char[strlen(s)+1];
            strcpy(p,s);
        }
        else {
            p = NULL;
        }
    }
    MyString (const MyString & s) {
        if( s.p ) {
            p = new char[strlen(s.p)+1];
            strcpy(p,s.p);
        }
        else {
            p = NULL;
        }
    }
    MyString & operator = (const MyString & s) {
        if( this == & s) {
            return * this;
        }
        if( s.p ) {
            p = new char[strlen(s.p)+1];
            strcpy(p,s.p);
        }
        else {
            p = NULL;
        }
    }
    MyString & operator = (const char * s) {
        if( p)
            delete [] p;
        if( s ) {
            p = new char[strlen(s)+1];
            strcpy(p,s);

        }
        else
            p = NULL;
    }
//your code ends here
```

```
};
int main()
{
    char w1[200],w2[100];
    while( cin >> w1 >> w2) {
        MyString s1(w1),s2 = s1;
        MyString s3(NULL);
        s3.Copy(w1);
        cout << s1 << "," << s2 << "," << s3 << endl;

        s2 = w2;
        s3 = s2;
        s1 = s3;
        cout << s1 << "," << s2 << "," << s3 << endl;

    }
}
```

****9531.cpp:

```
/*
程序填空

输入

多组数据，每组一行，整数 n
输出
对每组数据，输出一行，包括两个整数， n-5 和 n - 8

输入样例
20
30
输出样例
15,12
25,22


*/
#include <iostream>
using namespace std;
class MyInt
{
    int nVal;
    public:
    MyInt( int n) { nVal = n ;}
//your code starts here
    operator int() { return nVal;}
    MyInt & operator - (int i) {
        nVal -= i;
        return * this;
    }
//your code ends here
};
int Inc(int n) {
    return n + 1;
}
int main () {
    int n;
    while(cin >>n) {
        MyInt objInt(n);
        objInt-2-1-3;
        cout << Inc(objInt);
        cout <<″,″;
        objInt-2-1;
        cout << Inc(objInt) << endl;
    }
    return 0;
}
```

****9533.cpp:

```
/*
程序填空

输入
多组数据，每组两个整数
输出
对每组数据，输出一行，就是输入的两个整数

输入样例

2 3
4 5
输出样例
```

```
2,3
4,5
*/
#include <iostream>
using namespace std;
class Point {
    private:
        int x;
        int y;
    public:
        Point() { };
//your code starts here
        friend ostream &  operator << ( ostream & o, const Point & p) {
            cout << p.x << "," << p.y;
            return o;
        }
        friend istream & operator >> (istream & i,Point & p) {
            i >> p.x >> p.y;
            return i;
        }
//your code ends here
};
int main()
{
    Point p;
    while(cin >> p) {
        cout << p << endl;
     }
    return 0;
}
```

****9534.cpp:

```
/*
程序填空，输出指定结果

输入
多组数据，每组数据是两个非负整数s和 n。s最多可能200位， n用int能表示

输出
```

对每组数据，输出 6 行，内容分别是：

```
s+n
s+n
s+n
2n+1
2n+1
2n+2
```

样例输入

```
99999999999999999999999999888888888888888812345678901234567789 12
6 6
```

样例输出：

```
99999999999999999999999999888888888888888812345678901234567801
99999999999999999999999999888888888888888812345678901234567801
99999999999999999999999999888888888888888812345678901234567801
25
25
26
12
12
12
13
13
14
```

```
*/
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <cstdio>
using namespace std;
const int MAX = 110;
class CHugeInt {
//your code starts here
    private:
        char buf[220];
    public:
        void reverse(char * p) {
```

```
        int len = strlen(p);
        int i = 0,j = len -1;
        while(i <= j ) {
            swap(p[i],p[j]);
            ++i;
            --j;
        }
    }
    CHugeInt(char * p) {
        memset(buf,0,sizeof(buf));
        strcpy(buf,p);
        reverse(buf);
    }
    CHugeInt(int n) {
        memset(buf,0,sizeof(buf));
        sprintf(buf,″%d″,n);
        reverse(buf);
    }
    CHugeInt operator+(int n) {
        return * this + CHugeInt(n);
    }
    CHugeInt operator +(const CHugeInt & n) const  {
        CHugeInt tmp(0);
        int carry = 0;
        for(int i = 0;i < 210 ; ++i) {
            char c1 = buf[i];
            char c2 = n.buf[i];
            if( c1 == 0 && c2 == 0 && carry == 0)
                break;
            if( c1 == 0)
                c1 = ’0’;
            if( c2 == 0)
                c2 = ’0’;

            int k = c1 - ’0’ + c2 - ’0’ + carry;
            if( k >= 10) {
                carry = 1;
                tmp.buf[i] = k - 10 + ’0’;
            }
            else {
                carry = 0;
                tmp.buf[i] = k + ’0’;
            }
        }
```

```
            return tmp;
        }
        friend CHugeInt operator +(int n,const CHugeInt & h)
        {
            return h+n;
        }
        friend ostream &  operator <<(ostream & o,const CHugeInt & h) {
            int len = strlen(h.buf);
            for(int i = len -1 ; i >= 0; -- i)
                cout << h.buf[i];
            return o;
        }
        CHugeInt & operator += (int n) {
            * this = * this  + n;
            return * this;
        }
        CHugeInt & operator ++() {
            * this = * this + 1;
            return * this;
        }
        CHugeInt operator ++(int ) {
            CHugeInt tmp(*this);
            * this = tmp + 1;
            return tmp;
        }
//your code ends here
};
int  main()
{
    char s[210];
    int n;

    while (cin >> s >> n) {
        CHugeInt a(s);
        CHugeInt b(n);

        cout << a + b << endl;
        cout << n + a << endl;
        cout << a + n << endl;
        b += n;
        cout  << ++ b << endl;
        cout << b++ << endl;
        cout << b << endl;
    }
    return 0;
}
```