

源代码如下：

#include<iostream>

#include<cstdlib>

using namespace std;

void num\_rst(string& s) {

    while (s[0] == '0' && s.length() > 1) {

        s.erase(0, 1);

    }

}

int pos\_num\_comp(string s1, string s2) {

    if (s1.length() > s2.length())

        return 1;

    else if (s1.length() < s2.length())

        return -1;

    else if (s1 > s2)

        return 1;

    else if (s1 < s2)

        return -1;

    else

        return 0;

}

string Plus\_core(string s1, string s2) {//the result of pos+pos

    string str1, str2;

    if (s1.length() > s2.length()) {

        str1 = s1;

        str2 = s2;

    }

    else {

        str1 = s2;

        str2 = s1;

    }

    int len = str1.length() - str2.length();

    for (int i = str2.length() - 1; i >= 0; i--) {

        str1[i + len] += str2[i] - '0';

    }

    for (int i = str1.length() - 1; i > 0; i--) {

        while (str1[i] > '9') {

            str1[i] -= 10;

            str1[i - 1]++;

        }

    }

    if (str1[0] > '9') {

        str1 = "0" + str1;

        while (str1[1] > '9') {

            str1[1] -= 10;

            str1[0]++;

        }

    }

    return str1;

}

string Minus\_core(string s1, string s2) {//the result of pos-pos

    string str1, str2;

    switch (pos\_num\_comp(s1, s2)) {

        case 0:

            return "0";

            break;

        case 1:

            str1 = s1;

            str2 = s2;

            break;

        case -1:

            str1 = s2;

            str2 = s1;

            break;

        default:

            return "error\_Minus\_core";

    }

    int len = str1.length() - str2.length();

    for (int i = str2.length() - 1; i >= 0; i--) {

        str1[i + len] -= str2[i] - '0';

    }

    for (int i = str1.length() - 1; i > 0; i--) {

        while (str1[i] < '0') {

            str1[i] += 10;

            str1[i - 1]--;

        }

    }

    while (str1[0] == '0') {

        str1.erase(0, 1);

    }

    if (pos\_num\_comp(s1, s2) == -1)

        str1 = "-" + str1;

    return str1;

}

string Plus(string s1, string s2) {

    if (s1[0] == '-' && s2[0] == '-') {

        s1.erase(0, 1);

        s2.erase(0, 1);

        return "-" + Plus\_core(s1, s2);

    }

    if (s1[0] != '-' && s2[0] != '-')

        return Plus\_core(s1, s2);

    if (s1[0] == '-' && s2[0] != '-') {

        s1.erase(0, 1);

        return Minus\_core(s2, s1);

    }

    if (s1[0] != '-' && s2[0] == '-') {

        s2.erase(0, 1);

        return Minus\_core(s1, s2);

    }

    return "error\_Plus";

}

string Minus(string s1, string s2) {

    if (s1[0] == '-' && s2[0] == '-') {

        s1.erase(0, 1);

        s2.erase(0, 1);

        num\_rst(s1);

        num\_rst(s2);

        return Minus\_core(s2, s1);

    }

    if (s1[0] != '-' && s2[0] != '-') {

        num\_rst(s1);

        num\_rst(s2);

        return Minus\_core(s1, s2);

    }

    if (s1[0] == '-' && s2[0] != '-') {

        s1.erase(0, 1);

        num\_rst(s1);

        num\_rst(s2);

        return "-" + Plus\_core(s1, s2);

    }

    if (s1[0] != '-' && s2[0] == '-') {

        s2.erase(0, 1);

        num\_rst(s1);

        num\_rst(s2);

        return Plus\_core(s1, s2);

    }

    return "error\_Minus";

}

string multiply(string num1, string num2) {

    string str;

    bool pos\_neg = false;

    if (num1[0] == '-' && num2[0] != '-')

        pos\_neg = true;

    if (num1[0] != '-' && num2[0] == '-')

        pos\_neg = true;

    if (num1[0] == '-')

        num1.erase(0, 1);

    if (num2[0] == '-')

        num2.erase(0, 1);

    //��������0

    num\_rst(num1);

    num\_rst(num2);

    if (num1 == "0" || num2 == "0")

        return "0";

    if (num1.length() == 1 && num2.length() == 1) {

        num1[0] = (num1[0] - '0') \* (num2[0] - '0');

        if (num1[0] > 9) {

            num1 = "0" + num1;

            num1[0] += (num1[1]) / 10;

            num1[1] = (num1[1]) % 10 + '0';

        }

        if (num1.length() == 1)

            num1[0] += '0';

        while (num1[0] == '0') {

            num1.erase(0, 1);

        }

        if (pos\_neg)

            num1 = "-" + num1;

        return num1;

    }

    else {

        while (num1.length() > num2.length()) {

            num2 = "0" + num2;

        }

        while (num2.length() > num1.length()) {

            num1 = "0" + num1;

        }

        int len = num1.length();

        string  A = num1.substr(0, (len + 1) / 2),

            B = num1.substr((len + 1) / 2, len / 2),

            C = num2.substr(0, (len + 1) / 2),

            D = num2.substr((len + 1) / 2, len / 2);

        string  temp1 = multiply(A, C),

            temp2 = multiply(B, D),

            temp3 = multiply(Minus(A, B), Minus(D, C));

        string  str1 = temp1,

            str2 = temp2,

            str3 = temp2;

        for (int i = 0; i < len / 2 \* 2; i++) {

            str1 += "0";

        }

        str2 = Plus(str2,temp1);

        str2 = Plus(str2, temp3);

        for (int i = 0; i < len / 2; i++) {

            str2 += "0";

        }

        str1=Plus(str1,Plus(str2,str3));

        while (str1[0] == '0') {

            str1.erase(0, 1);

        }

        if (pos\_neg)

            str1 = "-" + str1;

        return str1;

    }

    return "error\_multiply";

}

int main()

{

    cout << "Insert two number a, b: " << endl;

    string a, b;

    cin >> a >> b;

    cout << "a+b=" << Plus(a, b) << endl;

    cout << "a-b=" << Minus(a, b) << endl;

    cout << "a\*b=" << multiply(a, b) << endl;

    return 0;

}