1002写出这个数

我的答案

```
#include<iostream>
using namespace std;
void translate(int sum)
{
    int a[4]{};
    int j=3;
    while (sum % 10 != 0)
    {
        a[j] = sum % 10;
        sum = sum / 10;
        j--;
    }
    j++; //复位

    int i;
    for (j; j < 4; j++)
    {
        i = a[j];
        if (i == 0)
            cout << "ling";
        else if (i == 1)
            cout << "yi";
        else if (i == 2)
            cout << "er";
        else if (i == 3)
            cout << "san";
        else if (i == 4)
            cout << "si";
        else if (i == 5)
            cout << "wu";
        else if (i == 6)
            cout << "liu";
        else if (i == 7)
            cout << "qi";
        else if (i == 8)
            cout << "ba";
        else
            cout << "jiu";
```

```
        if (j != 3)
            cout << " ";
    }
}
int main()
{
    int num,sum=0;
    int vnum = 0;
    //int prsum = 0;
    cin >> num;
    //求和
    while (num %10 !=0) {
        vnum = num / 10;
        sum = sum + (num % 10);
        num = vnum;
    };
   translate(sum);
    //cout << prsum << endl;
    return 0;
}
```

范例1（检查点1错误）

```
#include <stdio.h>
#include <string.h>
int main()
{
    char c[100];
    int a[5];
    char pinyin[][9] = {
"ling","yi","er","san","si","wu","liu","qi","ba","jiu"};
    int i,n,t;
    n = 0;
    t = 0;
    gets(c);
    for(i=0;i<strlen(c);i++)
        n = n + c[i] - 48;
    while (n!=0)
    {
        a[t] = n%10;
        n = n/10;
        t++;
    }
    printf("%s",pinyin[a[t-1]]);
    for(i=t-2;i>=0;i--)
        printf(" %s",pinyin[a[i]]);
}
```

范例2（正确）

```
/*
输入格式:
每个测试输入包含 1 个测试用例，即给出自然数 n 的值。这里保证 n 小于 10^•100••。
输出格式:
```

```
在一行内输出 n 的各位数字之和的每一位，拼音数字间有 1 空格，但一行中最后一个拼音数字后没有空格。
主要考察字符数组
*/

#include <stdio.h>
#include <string.h>  //字符串头文件
int main()
{
    char c[105];//位数，多留几位
    int a[5];//一百个9也就900
    char pinyin[][9] = {
"ling","yi","er","san","si","wu","liu","qi","ba","jiu"};
    int i,n,t;
    n = 0;
    t = 0;
    gets(c);
    for(i=0;i<strlen(c);i++)
        n = n + c[i] - 48;
        //或 n = n + c[i] - '0';
    //计算总数
    while (n!=0)
    {
        a[t] = n%10;
        n = n/10;
        t++;

        /*
         n = 0; t = 0;
        a[t] = n%10;
        n = n/10;
        t++;
        check
            n=235
            t    0   1   2  3
            a[i] 5   3   2
        */
    }//取每一位头数
    printf("%s",pinyin[a[t-1]]);//所以t-1
    for(i=t-2;i>=0;i--)
        printf(" %s",pinyin[a[i]]);//最后不要空格
}
```

1003 我要通过

```
#include<iostream>
using namespace std;
int main()
{
    int n;   //n<=10
    cin >> n;
    int ni = n;//一定在while循环前把n存下来！！！！！
    string mstr[15];  //最多10个字符串
    bool yon[15];     //对应字符串组的对错
    for (int i = 0; i < n; i++)
    {
        cin >> mstr[i];
        yon[i] = true;//初始全为正确
```

```
    }                //录入字符串组
    string perstr;   //把对字符串数组的操作变为单个字符串的操作
    while (n--) {    //对n个字符串执行n次
        perstr = mstr[n];
        int ip = -1, it = -1;//第一个p或t的位置
        for (int i = 0; i < perstr.length(); i++) {
            if (perstr[i] == 'P') {
                ip = i;
                break;
            }
        }            //找到第一个P
        for (int i = 0; i < perstr.length(); i++) {
            if (perstr[i] == 'T') {
                it = i;
                break;
            }
        }            //找到第一个T
        if (it < ip) {
            yon[n] = false;
            continue;
        }            //p只能在t前面
        for (int i = 0; i < perstr.length(); i++) {
            if (i == ip || i == it) {
                continue;
            }
            if (perstr[i] != 'A' && perstr[i] != ' ') {
                yon[n] = false;
                break;
            }
        }            //除唯一的P,T之外只能是‘A’和‘’
        if (yon[n] == false)
            continue;
        int ap = 0, aa = 0, at = 0;//统计PT前中后三个位置的A个数
        for (int i = 0; i < ip; i++) {
            if (perstr[i] == 'A')
                ap++;
        }
        for (int i = ip+1; i < it; i++) {
            if (perstr[i] == 'A')
                aa++;
        }
        for (int i = it+1; i < perstr.length(); i++) {
            if (perstr[i] == 'A')
                at++;
        }
        if (ap * aa != at || aa==0) { //前*中=后
            yon[n] = false;
        }
    }
    for (int i = 0; i < ni; i++)//结果输出
    {
        if (yon[i])
            cout << "YES" << endl;
        else
            cout << "NO" << endl;
    }
    return 0;
}
```

1014 福尔摩斯的约会

```
#include<iostream>
#include<cmath>
using namespace std;
int isal(char a) {
    if ((a >= 'a' && a <= 'z') || (a >= 'A' && a <= 'Z')) {
        return 1;
    }
    else {
        return 0;
    }
};
int main() {
    string s1, s2, s3, s4;
    //s1 = "1io";
    int flag = 0;
    string meek[7] = {"MON","TUE","WED","THU","FRI","SAT","SUN"};
    cin >> s1 >> s2 >> s3 >> s4;
    for (int i = 0; i < s1.length() && i < s2.length(); i++) {
        if (flag == 1) {
            if (s1[i] >= '0' && s1[i] <= '9' && s1[i] == s2[i]) {
                cout << '0'<<s1[i] << ':';
                break;
            }
            if (s1[i] >= 'A' && s1[i] <= 'N' && s1[i] == s2[i]) {
                cout << 10 + (s1[i] - 'A') << ':';
                break;
            }
        }
        else if (s1[i] >= 'A' && s1[i] <= 'G' && s1[i] == s2[i] ) {
            flag = 1;
            cout << meek[s1[i] - 'A'] << " ";
            //字符之间做减法
        }
    }
    for (int i = 0; i < s3.length() && i < s4.length(); i++) {
        if (s3[i] == s4[i] && i<10 && isal(s3[i])) {
            cout << '0' <<i;
            break;
        }
        else if(s3[i] == s4[i] && isal(s3[i])){
            cout << i;
            break;
        }
    }
    return 0;
}
```

15年武汉第四题

1015德才论

//写了一个小时还没有完全通过。。。。

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

class stu {
public:
    long long int num;
    int de;
    int cai;
    int sum; //德+才
};
stu  kao[100000];
class paixu {
public:
    int ik;//第i个考生
    int sum;//考生成绩
};
paixu one[1000];
paixu two[1000];
paixu thr[1000];
paixu fou[1000];

int main()
{
    //n考生总数
    //l最低分数线
    //h优先录取线
    int n, l, h;
    int count = 0;//达标人的总数
    int i1 = 0, i2 = 0, i3 = 0, i4 = 0;//每一类的人数
    cin >> n >> l >> h;
    for (int i = 0; i < n; i++) {
        cin >> kao[i].num >> kao[i].de >> kao[i].cai;
        kao[i].sum = kao[i].de + kao[i].cai;
    }
    for (int i = 0; i < n; i++) {
        if (kao[i].de >= h && kao[i].cai >= h) {
            one[i1].ik = i;
            one[i1].sum = kao[i].sum;
            i1++;
            count++;
        }
        else if (kao[i].de >= h && kao[i].cai < h && kao[i].cai >= l) {
            two[i2].ik = i;
            two[i2].sum = kao[i].sum;
            i2++;
            count++;
        }
```

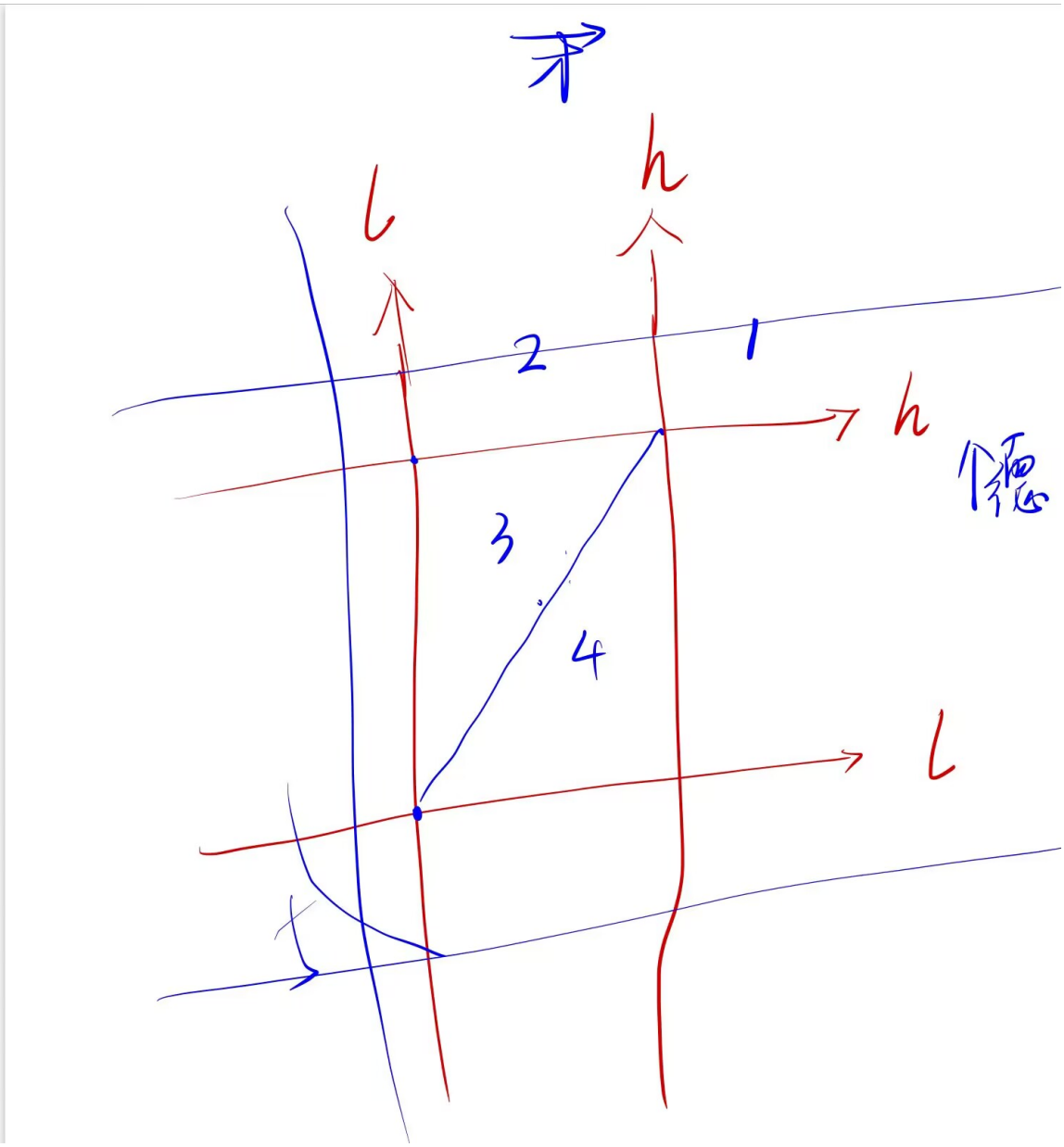
```
        else if (kao[i].cai < h && kao[i].cai >= l && kao[i].de < h && kao[i].de
>= l && kao[i].de >= kao[i].cai) {
            thr[i3].ik = i;
            thr[i3].sum = kao[i].sum;
            i3++;
            count++;
        }
        else if (kao[i].cai >= l && kao[i].de >= l) {
            fou[i4].ik = i;
            fou[i4].sum = kao[i].sum;
            i4++;
            count++;
        }
    }
    for (int j = 0; j < i1; j++) {
        int max=0,maxc=0;
        paixu tmp;
        for (int k = j; k < i1; k++) {
            if (one[k].sum > max) {
                max = one[k].sum;
                maxc = k;
            }
            if (one[k].sum == max) {
                if (kao[one[k].ik].de > kao[one[maxc].ik].de) {
                    max = one[k].sum;
                    maxc = k;
                }
                if (kao[one[k].ik].de == kao[one[maxc].ik].de) {
                    if (kao[one[k].ik].num < kao[one[maxc].ik].num) {
                        max = one[k].sum;
                        maxc = k;
                    }
                }
            }
        }
        tmp = one[maxc];
        one[maxc] = one[j];
        one[j] = tmp;
    }
    for (int j = 0; j < i2; j++) {
        int max = 0, maxc = 0;
        paixu tmp;
        for (int k = j; k < i2; k++) {
            if (two[k].sum > max) {
                max = two[k].sum;
                maxc = k;
            }
            if (two[k].sum == max) {
                if (kao[two[k].ik].de > kao[two[maxc].ik].de) {
                    max = two[k].sum;
                    maxc = k;
                }
                if (kao[two[k].ik].de == kao[two[maxc].ik].de) {
                    if (kao[two[k].ik].num < kao[two[maxc].ik].num) {
                        max = two[k].sum;
                        maxc = k;
                    }
                }
```

```
            }
        }
        tmp = two[maxc];
        two[maxc] = two[j];
        two[j] = tmp;
    }
    for (int j = 0; j < i3; j++) {
        int max = 0, maxc = 0;
        paixu tmp;
        for (int k = j; k < i3; k++) {
            if (thr[k].sum > max) {
                max = thr[k].sum;
                maxc = k;
            }
            if (thr[k].sum == max) {
                if (kao[thr[k].ik].de > kao[thr[maxc].ik].de) {
                    max = thr[k].sum;
                    maxc = k;
                }
                if (kao[thr[k].ik].de == kao[thr[maxc].ik].de) {
                    if (kao[thr[k].ik].num < kao[thr[maxc].ik].num) {
                        max = thr[k].sum;
                        maxc = k;
                    }
                }
            }
        }
        tmp = thr[maxc];
        thr[maxc] = thr[j];
        thr[j] = tmp;
    }
    for (int j = 0; j < i4; j++) {
        int max = 0, maxc = 0;
        paixu tmp;
        for (int k = j; k < i4; k++) {
            if (fou[k].sum > max) {
                max = fou[k].sum;
                maxc = k;
            }
            if (fou[k].sum == max) {
                if (kao[fou[k].ik].de > kao[fou[maxc].ik].de) {
                    max = fou[k].sum;
                    maxc = k;
                }
                if (kao[fou[k].ik].de == kao[fou[maxc].ik].de) {
                    if (kao[fou[k].ik].num < kao[fou[maxc].ik].num) {
                        max = fou[k].sum;
                        maxc = k;
                    }
                }
            }
        }
        tmp = fou[maxc];
        fou[maxc] = fou[j];
        fou[j] = tmp;
    }

    cout << count << endl;
```

```
    for (int j = 0; j < i1; j++) {
        cout << kao[one[j].ik].num <<' ' << kao[one[j].ik].de << ' ' <<
kao[one[j].ik].cai << endl;
    }
    for (int j = 0; j < i2; j++) {
        cout << kao[two[j].ik].num << ' ' << kao[two[j].ik].de <<' ' <<
kao[two[j].ik].cai << endl;
    }
    for (int j = 0; j < i3; j++) {
        cout << kao[thr[j].ik].num << ' ' << kao[thr[j].ik].de << ' ' <<
kao[thr[j].ik].cai << endl;
    }
    for (int j = 0; j < i4; j++) {
        cout << kao[fou[j].ik].num << ' ' << kao[fou[j].ik].de << ' ' <<
kao[fou[j].ik].cai << endl;
    }
    return 0;
}
```

看解析之后的ac版本




```
#include<iostream>
```

```cpp
#include<algorithm>
using namespace std;

class stu {
public:
    long long int num; //学号
    int de, cai, sum;  //德，才，总分
    int lei;           //类型1 2 3 4 5
};
stu s[100001];
bool tmp(stu a,stu b) {   //排序
    if (a.lei != b.lei)
    {
        return a.lei < b.lei; //小的放在了前面
    }
    else if (a.sum != b.sum) {
        return a.sum > b.sum; //高分在前面
    }
    else if (a.de != b.de) {
        return a.de > b.de;
    }
    else {
        return a.num < b.num;
    }
};
int main() {
    int n, l, h;        //考生总数，最低线，优秀线
    int ans = 0;        //达标的考生总数
    cin >> n >> l >> h;
    for (int i = 0; i < n; i++) {
        cin >> s[i].num >> s[i].de >> s[i].cai;
        s[i].sum = s[i].de + s[i].cai;
    }
    for (int i = 0; i < n; i++) {      //考生分类
        if (s[i].de < l || s[i].cai < l) {
            s[i].lei = 5;
        }
        else if (s[i].de >= h && s[i].cai >= h) {
            s[i].lei = 1;
            ans++;
        }
        else if (s[i].de >= h && s[i].cai >= l) {
            s[i].lei = 2;
            ans++;
        }
        else if (s[i].de >= l && s[i].cai >= l && s[i].de >= s[i].cai) {
            s[i].lei = 3;
            ans++;
        }
        else {
            s[i].lei = 4;
            ans++;
        }
    }
    sort(s, s + n, tmp);
    cout << ans << endl;
    for (int i = 0; i < ans; i++) {
        cout << s[i].num << ' ' << s[i].de << ' ' << s[i].cai << endl;
```

```
    }
    return 0;
}
```

1018 石头剪刀布

//没有完全AC

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

int main() {
    long long int n;
    char a, b;   //a代表甲 b代表乙
    int jia = 0, yi = 0, ping = 0;
    char shou[3] = { 'J','B','C' };
    int as[3] = { 0 }, bs[3] = { 0 };//0J 1B 2C  ab对应手势胜利次数
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a >> b;
        if (a == b) {
            ping++;
        }
        else if (a == 'C' && b == 'J') {
            jia++;
            as[2]++;
        }
        else if (a == 'J' && b == 'B') {
            jia++;
            as[0]++;
        }
        else if (a == 'B' && b == 'C') {
            jia++;
            as[1]++;
        }
        else if (b == 'C' && a == 'J') {
            yi++;
            bs[2]++;
        }
        else if (a == 'J' && b == 'B') {
            yi++;
            bs[0]++;
        }
        else {
            yi++;
            bs[1]++;
        }
    }
    cout << jia << ' ' << ping << ' ' << yi << endl;
    cout << yi << ' ' << ping << ' ' << jia << endl;
    int max = 0;
    char maxa,maxb;
    for (int i = 0; i < 3; i++) {
        if (as[i] > max) {
            maxa = shou[i];
            max = as[i];
        }
```

```
        else if (as[i] == max && shou[i] < shou[max]) {
            maxa = shou[i];
        }
    }
    if (max == 0) {
        maxa = 'B';
    }
    max = 0;
    for (int i = 0; i < 3; i++) {
        if (bs[i] > max) {
            maxb = shou[i];
            max = bs[i];
        }
        else if (bs[i] == max && shou[i] < shou[max]) {
            maxb = shou[i];
        }
    }
    if (max == 0) {
        maxb = 'B';
    }
    cout << maxa << ' ' << maxb;
    return 0;
}
```

1019数字黑洞

第一次AC的代码

```
#include<iostream>
using namespace std;
int f(int n) {   //非递增排序
    int a[4] = { 0 };   //寄存n的4位
    int b[4] = { 0 };
    int max, min;//对应a,b
    for (int i = 0; i < 4; i++) {
        a[i] = n % 10;
        b[i] = a[i];
        n = n / 10;
    }
    int tmp;
    for(int i=0;i<4;i++)
        for (int j = i; j < 4; j++) {
            if (a[j] > a[i]) {
                tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
            }
        }
    for (int i = 0; i < 4; i++)
        for (int j = i; j < 4; j++) {
            if (b[j] < b[i]) {
                tmp = b[i];
                b[i] = b[j];
                b[j] = tmp;
```

```
            }
        }
    for (int i = 0; i < 4; i++) {
        cout << a[i];
    }
    cout << " - ";
    for (int i = 0; i < 4; i++) {
        cout << b[i];
    }
    max = a[0] * 1000 + a[1] * 100 + a[2] * 10 + a[3];
    min = b[0] * 1000 + b[1] * 100 + b[2] * 10 + b[3];
    /*cout << " = " << max - min;*/
    return max - min;
};
bool ff(int n) {//判断是否4位全等
    if (n >= 1111 && n % 1111 == 0) {
        return true;
    }
    else {
        return false;
    }
};
int main() {
    int n; //n<10000
    int max,min;
    cin >> n;
    do {
        if (ff(n))
        {
            //cout << "N - N = 0000";
            cout << n << " - " << n << " = 0000";
            break;
        }
        else {
            n = f(n);
            int ln = n;  //临时的n
            if (n > 1000) {
                cout << " = " << n << endl;
            }
            else {
                int a[4] = { 0 };
                for (int i = 3; i >=0; i--) {
                    a[i] = ln % 10;
                    ln = ln / 10;
                }
                cout << " = ";
                for (int i = 0; i < 4; i++) {
                    cout << a[i];
                }
                cout << endl;
            }

        }


    } while (n != 6174);
    return 0;
}
```

改进后的代码1.0（用了sort函数）

```
#include<iostream>
#include<algorithm>
using namespace std;
bool cmp(int a, int b) {
    return a > b;
}
bool cmp1(int a, int b) {
    return a < b;
}
int f(int n) {   //非递增排序
    int a[4] = { 0 };   //寄存n的4位
    int b[4] = { 0 };
    int max, min;//对应a,b
    for (int i = 0; i < 4; i++) {
        a[i] = n % 10;
        b[i] = a[i];
        n = n / 10;
    }
    sort(a, a + 4, cmp);
    sort(b, b + 4, cmp1);
    for (int i = 0; i < 4; i++) {
        cout << a[i];
    }
    cout << " - ";
    for (int i = 0; i < 4; i++) {
        cout << b[i];
    }
    max = a[0] * 1000 + a[1] * 100 + a[2] * 10 + a[3];
    min = b[0] * 1000 + b[1] * 100 + b[2] * 10 + b[3];
    /*cout << " = " << max - min;*/
    return max - min;
};
int main() {
    int n; //n<10000
    int max,min;
        cin >> n;
        do {
            n = f(n);
            int ln = n;  //临时的n
            if (n > 1000) {
                cout << " = " << n << endl;
            }
            else {
                int a[4] = { 0 };
                for (int i = 3; i >= 0; i--) {
                    a[i] = ln % 10;
                    ln = ln / 10;
                }
                cout << " = ";
                for (int i = 0; i < 4; i++) {
                    cout << a[i];
```

```
                }
                cout << endl;
            }
        } while (n != 6174 && n != 0);
    return 0;
}
```

改进后的代码2.0（加入insert函数，对字符串操作）

```
#include<iostream>
#include<algorithm>

using namespace std;
bool cmpa(char a, char b) {
    return a > b;
};
bool cmpb(char a, char b) {
    return a < b;
};
int main() {
    string n;
    string a, b;
    int nn, ma , mb;
    cin >> n;
    n.insert(0, 4 - n.length(), '0'); //在0号下标前加入 4-长 个0
    do {
        a = n; b = n;
        ma = 0; mb = 0;
        sort(a.begin(), a.end(), cmpa);//降序
        sort(b.begin(), b.end(), cmpb);//升序
        for (int i = 0; i < 4; i++) {
            int m = a[i] - '0';//某一位对应数字
            ma = ma * 10 + m;
        }
        for (int i = 0; i < 4; i++) {
            int m = b[i] - '0';//某一位对应数字
            mb = mb * 10 + m;
        }
        nn = ma - mb;           //对应数字n
        n = "";
        for (int i = 0; i < 4; i++) {
            char q = (nn % 10) + '0';
            n = q + n;
            nn = nn / 10;
        }
        n.insert(0, 4 - n.length(), '0');
        cout << a << " - " << b << " = " << n << endl;
    } while (n != "0000" && n != "6174");

    return 0;
}
```