



# 计算机高级语言课程设计报告

学生姓名： 徐鸿飞

班 学 号： 115172——20171002608

指导教师： 左泽均 杨林权 赵一石

中国地质大学信息工程学院

2017 年 12 月 28 日

# 实习题目一

## 【需求规格说明】

1、观察下面的加法算式：

祥瑞生辉  
+ 三羊献瑞  
-----

三羊生瑞气

其中，相同的汉字代表相同的数字，不同的汉字代表不同的数字。请你编程计算“三羊献瑞”所代表的 4 位数字。

问题描述：要输出“三羊献瑞”所代表的数字。

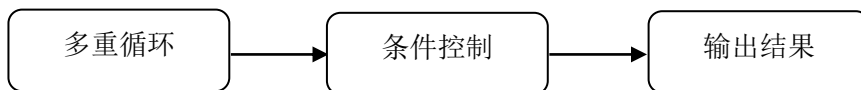
## 【算法设计】

(1) 设计思想：

先观察题目，根据所学的数学知识知道其中“三”只能等于 1，然后其他汉字用变量表示，根据算式上的关系，利用循环条件语句得出答案。

(2) 设计表示：

分别用字母变量表示除“三”之外的其他汉字，然后多重循环，再用条件语句得出结果。



## 【调试报告】

刚开始的程序可以运行，但是得出的结果是错误的，经过仔细检查之后发现是由于看错了题目，没有发现“祥瑞生辉”和“三羊献瑞”中重复的“瑞”字，所以输出了多组结果。经改正之后得出了正确答案。但是仍有不足之处：控制语句太长。

## 【附录】

源程序清单：


```
#include "stdafx.h"
#include<iostream>
using namespace std;
int main()
{
    int a, b, c, d, e, f, g, h;
    for (a = 0; a <= 9; a++) //多重循环
        for (b = 0; b <= 9; b++)
            for (c = 0; c <= 9; c++)
                for (d = 0; d <= 9; d++)
                    for (e = 1; e <= 9; e++)
                        for (f = 0; f <= 9; f++)
                            for (g = 0; g <= 9; g++)
                                for (h = 0; h <= 9; h++)
                                    if (a * 1000 + b * 100 + c * 10 + d + e * 1000 + f
```

```

* 100 + g * 10 + b == e * 10000 + f * 1000 + c * 100 + b * 10 + h&& a != b && a != c &&
a != d && a != e&& a != f && a != g && a != h&& b != c && b != d && b != e&& b != f && b !=
g && b != h&& c != d && c != e&& c != f && c != g && c != h&& d != e&& d != f && d != g && d !=
h&& f != g && f != h&& g != h&& a != 1 && b != 1 && c != 1 && d != 1 && f != 1 && g !=
1 && h != 1 && e == 1) //条件控制
    cout << "三=" << e << '\t' << "羊=" << f << '\t' << "献=" << g << '\t' <<
"瑞=" << b << endl; //输出结果
    ::system("pause"); //暂停
    return 0;
}

```

运行结果:

 选择C:\Users\徐鸿飞\Desktop\课程设计\三羊献瑞\Debug\三羊献瑞.exe

三=1    羊=0    献=8    瑞=5  
 请按任意键继续. . .

## 实习题目二

### 【需求规格说明】

3、A、B、C、D、E 五名学生有可能参加计算机竞赛，根据下列条件判断哪些人参加了竞赛：

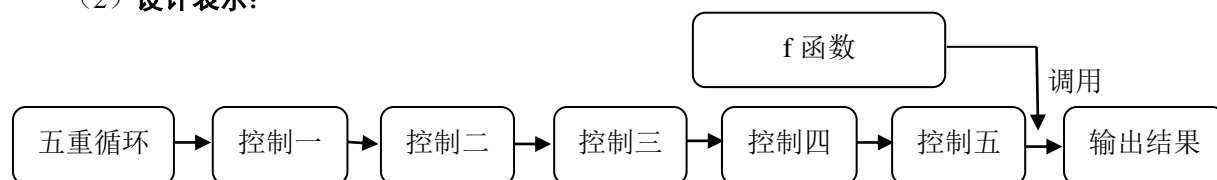
- (1) A 参加时，B 也参加；
- (2) B 和 C 只有一个人参加；
- (3) C 和 D 或者都参加，或者都不参加；
- (4) D 和 E 中至少有一个人参加；
- (5) 如果 E 参加，那么 A 和 D 也都参加。

最终输出结果。

### 【算法设计】

(1) **设计思想：**分别用“0”和“1”来表示“不参加”和“参加”，然后分别将 A、B、C、D、E 定义为五个变量在 0~1 之间循环，再将题中的五个控制条件用逻辑语句表示，控制循环的进行。编写一个 f 函数，将“0”和“1”还原成“不参加”和“参加”的结果输出。

(2) **设计表示：**



### 【调试报告】

本题比较简单，没有出现问题。

### 【附录】

源程序清单:

```

#include "stdafx.h"
#include<iostream>
using namespace std;
char *f(int n) //f函数将“0”和“1”还原成“不参加”和“参加”

```

```

{
    if (n == 0) return "不参加";
    if (n == 1) return "参加";
}

int main()
{
    int a, b, c, d, e;
    for (a = 0; a <= 1; a++) //多重循环
        for (b = 0; b <= 1; b++)
            for (c = 0; c <= 1; c++)
                for (d = 0; d <= 1; d++)
                    for (e = 0; e <= 1; e++)
                    {
                        if (a&&!b)
                            continue;
                        if (b==c)
                            continue;
                        if (c!=d)
                            continue;
                        if (!d&&!e)
                            continue;
                        if (e && (!a || !d))
                            continue; //条件控制
                        cout << 'A' << f(a)<<'\\n'<< 'B' << f(b) << '\\n' << 'C' << f(c) << '\\n' << 'D' << f(d) << '\\n' << 'E' << f(e) << endl; //输出结果调用f函数
                    }
                ::system("pause");
    return 0;
}

```

运行结果：



## 实习题目三

### 【需求规格说明】

A-4、10 人围坐成一圈（假设他们的编号沿顺时针方向依次为 1 到 10）。编程序，使用数组来存放各数据（人员编号），从 1 号人员开始数起（沿顺时针方向），当数到 k 时（其中 k>1，由用户通过 cin 输入指定），则该号人员被“淘汰出局”；接着仍沿顺时针方向从被淘汰出局者

的下一人员又 重新从 1 开始数起，数到 k 后，淘汰第 2 个人；如此继续，直到最后剩下一个人时停止。请输出 最后所剩那一个人的编号，并输出淘汰过程的某种“中间结果数据”。

如已知：1 2 3 4 5 6 7 8 9 10, k=5;

第一次：1 2 3 4 0 6 7 8 9 10, 淘汰出局：5;

第二次：1 2 3 4 0 6 7 8 9 0, 淘汰出局：10;

第三次：1 2 3 4 0 0 7 8 9 0, 淘汰出局：6;

第四次：1 0 3 4 0 0 7 8 9 0, 淘汰出局：2;

第五次：1 0 3 4 0 0 7 8 0 0, 淘汰出局：9;

第六次：1 0 3 4 0 0 7 0 0 0, 淘汰出局：8;

第七次：0 0 3 4 0 0 7 0 0 0, 淘汰出局：1;

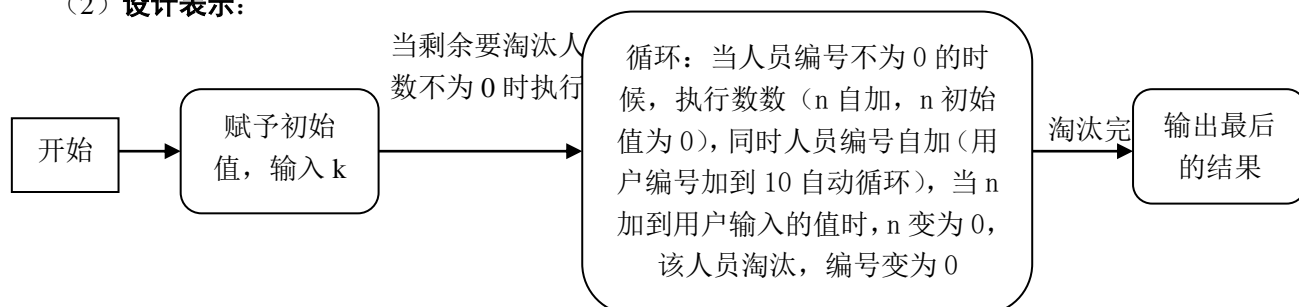
第八次：0 0 3 0 0 0 7 0 0 0, 淘汰出局：4;

第九次：0 0 3 0 0 0 0 0 0 0, 淘汰出局：7。

### 【算法设计】

(1) **设计思想**：由题意可以知道一共有 10 个人，分别给他们编号为 1~10（数组表示），要淘汰 9 个人，淘汰的人将他的编号变为 0。设置循环：m=9，当 m 不为 0 的时候执行循环，每淘汰一个人，m 自减。循环里嵌套选择条件，当人员编号不为 0 的时候，执行数数（n 自加，n 初始值为 0），同时人员编号自加（用户编号加到 10 自动循环），当 n 加到用户输入的值时，n 变为 0，该人员淘汰，编号变为 0，不再参与数数的环节。当淘汰九个人以后，m=0，循环结束。输出最后剩下人员的编号。

(2) **设计表示**：



### 【调试报告】

刚开始程序可以运行，但是运行的非常的慢，而且有数据错误，经排查发现是初始值设置错误，且嵌套多个循环，经改正之后正确。

### 【附录】

源程序清单：

```
#include "stdafx.h"
#include<iostream>
using namespace std;
int main()
{
int i, k, m, n, p[10];
cin>>k;
for(i=0; i<10; i++)
{p[i]=i+1;} // 给十个人分别编号
i=0;
```

---

```

n=0;//n为a为要数的数变量
m=9;// 共淘汰9个人
while(m)
{
    if(p[i]!=0)
        n++;//当该人员编号不为0时，数数
    if(n==k)// 当数到用户输入的值后，淘汰该人员，同时n变为0，重新开始数数，要淘汰的人
数减1
    {
        cout<<"淘汰："<<p[i]<<' \t'<<"剩余情况： ";
        p[i]=0;
        n=0;
        for(int j=0;j<10;j++)
            {cout<<" "<<p[j];}
        cout<<endl;
        m--;
    }
    i++;
    if(i==10)    {i=0;}
}
for(int a=0;a<10;a++)
{
    if(p[a]==0){continue;}// 最后剩下的人编号不为0
    else    cout<<"最后剩的人的编号是："<<p[a]<<endl;
}

::system("pause");
return 0;
}

```

**测试数据：**

5

**运行结果：**

```
C:\Users\SXY\Desktop\课程设计\淘汰\Debug\淘汰.exe
5
淘汰: 5 剩余情况: 1 2 3 4 0 6 7 8 9 10
淘汰: 10 剩余情况: 1 2 3 4 0 6 7 8 9 0
淘汰: 6 剩余情况: 1 2 3 4 0 0 7 8 9 0
淘汰: 2 剩余情况: 1 0 3 4 0 0 7 8 9 0
淘汰: 9 剩余情况: 1 0 3 4 0 0 7 8 0 0
淘汰: 8 剩余情况: 1 0 3 4 0 0 7 0 0 0
淘汰: 1 剩余情况: 0 0 3 4 0 0 7 0 0 0
淘汰: 4 剩余情况: 0 0 3 0 0 0 7 0 0 0
淘汰: 7 剩余情况: 0 0 3 0 0 0 0 0 0 0
最后剩的人的编号是: 3
请按任意键继续. . .
```

## 实习题目四

### 【需求规格说明】

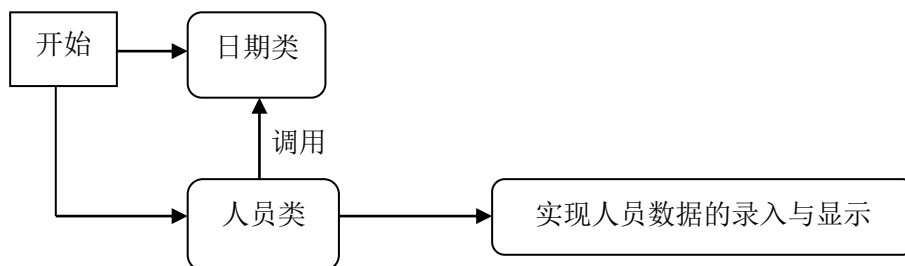
B-7、设计一个用于人事管理的“人员”类。由于考虑到通用性，这里只抽象出所有类型人员都具有的属性：编号、性别、出生日期、身份证号等。其中“出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、复制构造函数、带默认形参值的成员函数、类的组合。

问题描述：创建类，在主函数里调用类中的成员函数实现对人员信息的录入和显示。

### 【算法设计】

(1) **设计思想：**先写一个日期类实现日期的输入与输出，然后再写一个人员类，可以在人员类里访问日期类里的函数，实现各数据的录入与输出。

(2) **设计表示：**



### 【调试报告】

刚开始写的程序本来没有问题，但是没有加入析构函数，没有做好“清理善后”的工作，我认识到这样是不好的，所以之后加入了析构函数。

### 【附录】

源程序清单：

```
#include "stdafx.h"
#include<iostream>
#include <string>
using namespace std;
class Date //日期类
{
private:
```

---

```

    int year;
    int month;
    int day;
public: Date() {} //默认构造
    Date(int y, int m, int d) //带参构造
    { year=y; month=m; day=d; }
    void set() //设置数据函数
    { cin>>year>>month>>day; }
    void display() //显示函数
    { cout<<year<<"年" <<month<<"月?"<<day<<"日" <<"; } };

class Person //人员类
{
private:
    int num; //编号
    char sex;
    Date birthday;
    char ID[18]; //身份证号
public: Person() {} //默认构造
    Person(int n, char id[18], char s='m')
    { num=n; sex=s; strcpy(ID, id); } //有默认值的带参构造
    Person(Person& p) //拷贝构造
    { num=p.num; sex=p.sex; birthday=p.birthday; strcpy(ID, p.ID); }
    void input() //输入函数
    { cout<<"录入数据: " <<endl;
        cout<<"编号: ";
        cin>>num;
        cout<<"性别(m/w): ";
        cin>>sex; cout<<"生日": ";
        birthday.set();
        cout<<"身份证号: ";
        cin>>ID; ID[18]='\0';
        cout<<endl; }
    void output() //输出函数
    { cout<<"编号: " <<num<<endl;
        cout<<"性别: ";
        if(sex=='m')
        {cout<<"男" <<endl;}
        else if(sex=='w')
        {cout<<"女";}
        else {cout<<"输入错误! ";}
        cout<<"生日: "; birthday.display();

```



```

        cout<<endl; cout<<"身份证号: "<<ID<<endl; }
~Person() //析构函数
{ cout<<" "<<num<<"号人员已经录入"<<endl; } };

```

```

int main()
{
    Person p1; //定义类的对象
    p1.input(); //调用类中输入函数
    p1.output(); //调用类中输出函数
    ::system("pause");
    return 0;
}

```

运行结果:

```

C:\Users\SIT\Desktop\课程设计\test\Debug\test.exe
录入数据:
编号: 1
性别<m/w>: m
生日: 1111 1 1
身份证号: 111111111111111111

编号: 1
性别: 男
生日: 1111年1月1日
身份证号: 111111111111111111
请按任意键继续. . .

```

## 实习题目五

### 【需求规格说明】

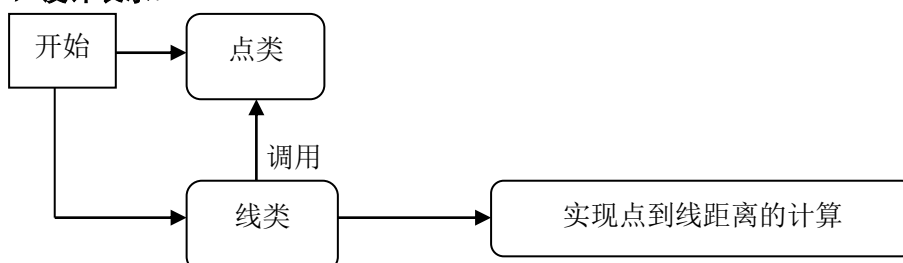
B-8、计算点到折线的距离。

点到折线的距离定义为点到折线中每一条线段的距离中的最短距离，而点到线段的距离可能是点到线段某个端点的距离（当垂足不在线段内部），也可能是点到垂足的距离（当垂足在线段内部）。要求定义点类 CPoint 和折线类 CLine，并在折线类上实现距离计算成员函数。

### 【算法设计】

**(1) 设计思想:** 先写一个点类定义点，然后写一个线类，并在线类里定义一个点对象，在线类里面添加计算距离的成员函数。点用坐标 (x0, y0) 表示，折线用  $ax+by+c=0$  表示，然后用公式  $d=|ax_0+by_0+c|/\sqrt{a^2+b^2}$  计算点到直线的距离。

**(2) 设计表示:**



---

## 【调试报告】

本题比较简单，没有出现问题。

## 【附录】

```
#include "stdafx.h"
#include<iostream>
#include<cmath>
using namespace std;
class cpoint//点类
{
private:
    double x,y;
public:
    cpoint(double xx=0, double yy=0)
    {x=xx;y=yy;}
    double get_x() {return x;}
    double get_y() {return y;}
    void set()
    {
        cout<<"请输入点坐标"<<endl;
        cin>>x>>y;
    }
};
class cline//折线类
{
private:
    cpoint poi;//定义点类
    double a,b,c;//折线: ax+by+c=0
    double x0,y0;
    double d;
public:
    cline() {}//默认构造
    cline(double _a, double _b, double _c)
    {a=_a;b=_b;c=_c;}
    void input()//输入函数°
    {
        poi.set();
        cout<<"输入折线系数"<<endl;
        cin>>a>>b>>c;}
    void fun()//值传递
    {x0=poi.get_x();
```

```

        y0=poi.get_y();}
double fun1()
{
double s=fabs(a*x0+b*x0+c);
double m=sqrt(a*a+b*b);
d=s/m;
return d;
}
void output() //输出函数
{cout<<"点到直线距离为a: " <<d<<endl;}
~cline() {} //析构函数
};

int main()
{
    cline c1;
    c1.input();
    c1.fun();
    c1.fun1();
    c1.output();
    ::system("pause");
    return 0;
}

```

测试数据:

点 (2,3) 线:  $x+3y=2$

运行结果:



## 实习题目六

### 【需求规格说明】

B-9、计算三角形面积:

### 【问题描述】

给出三角形的三个边长为  $a$ ,  $b$ ,  $c$ , 根据海伦公式来计算三角形的面积:

$s = (a+b+c)/2;$

$area = \text{sqrt}(s*(s-a)*(s-b)*(s-c));$

### 【要求】

**【数据输入】** 测试的数据有任意多组, 每一组为一行。

每一行为三角形的三个边长为  $a$ ,  $b$ ,  $c$ ;

**【数据输出】** 输出每一个三角形的面积, 两位小数。如果不是一个三角形, 则输出错误提示信息: "Input error!"

### 【样例输入】

3 4 5

6 8 10

1 2 3

### 【样例输出】

6. 00

24. 00

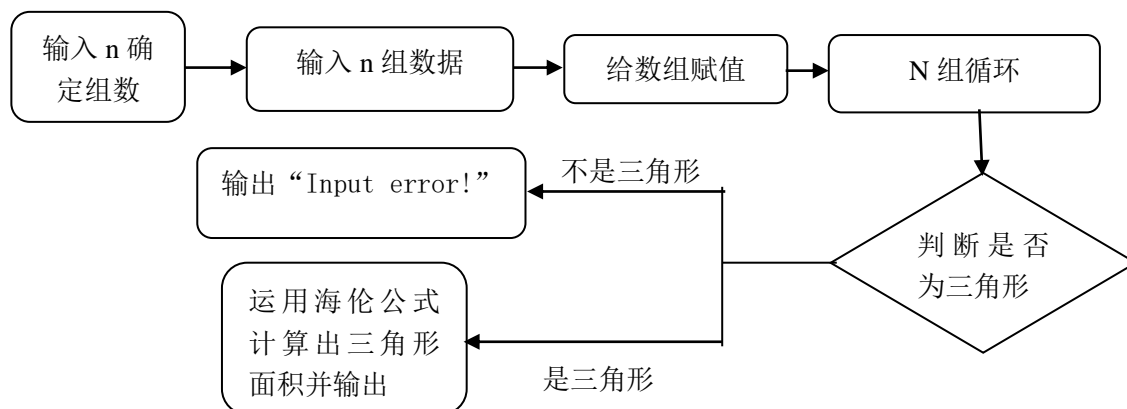
Input error!

问题描述: 要求输入多组三角形的面积, 每组为一行, 然后直接输出所有组数的三角形的面积。

### 【算法设计】

(1) **设计思想:** 由于要输入多组数据, 且每组为一行, 所以可以先输入一个  $n$  来确定要输入的组数, 然后创建数组赋值, 再用循环计算出每组三角形的面积。

(2) **设计表示:**



### 【调试报告】

刚开始写的程序中没有想到先输入  $n$  来控制组数，所以没有实现每组为一行，而是每次输入之后回车，面积就直接显示了，而不是集中显示的，后来经过修改，虽然增加了  $n$ ，但是达到了题目的要求。

### 【附录】

源程序清单：

```
#include "stdafx.h"
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;
int main()
{
    double a[100], b[100], c[100]; //定义数组
    double s, n;
    float area;
    cout << "请输入要测试的数据的组数" << endl;
    cin >> n; //共输入n组数据
    cout << "请输入三角形三条边长：" << endl;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i] >> b[i] >> c[i]; //将输入的值赋给数组
    }
    for (int i = 1; i <= n; i++) //循环判断计算
    {
        s = (a[i] + b[i] + c[i]) / 2;
        (float)area = sqrt(double(s*(s - a[i])*(s - b[i])*(s - c[i])));
        if (a[i] + b[i]>c[i] && a[i] + c[i]>b[i] && b[i] + c[i]>a[i]) //判断是否为三角形
        {
            cout << setprecision(2) << fixed /*控制小数位数*/ << "该三角形面积为：" <<
            area << endl; //是三角形则输出面积
        }
        else
        {
            cout << "Input error!" << endl; //不是三角形则输出Input error!
        }
    }
    ::system("pause"); //暂停
    return 0;
}
```

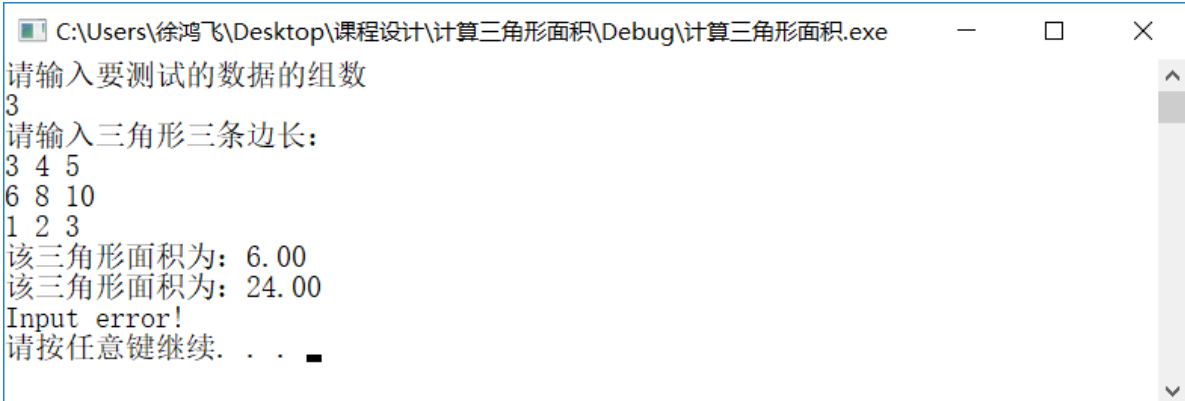
测试数据：

3 4 5

6 8 10

1 2 3

运行结果:



```
C:\Users\徐鸿飞\Desktop\课程设计\计算三角形面积\Debug\计算三角形面积.exe
请输入要测试的数据的组数
3
请输入三角形三条边长:
3 4 5
6 8 10
1 2 3
该三角形面积为: 6.00
该三角形面积为: 24.00
Input error!
请按任意键继续. . .
```

## 实习题目七

### 【需求规格说明】

#### 飞机加油问题

Time Limit:2s Memory limit:32M

Accepted Submit:136 Total Submit:934

#### 【问题描述】

F 国际航空公司在世界范围有  $n$  个国际机场。第  $i$  个国际机场到中心机场的距离为  $d_i$ ,  $i=1, \dots, n$ 。从国际机场  $j$  到国际机场  $i$  的飞行费用为  $c(i,j) = s + (d_j - d_i)^2$ ,  $s$  为地面加油费用。从任何国际机场飞往中心机场的飞机可以在任一国际机场加油后继续飞行。飞机加油问题要求确定从距中心机场最远的国际机场飞到中心机场的最少费用。

#### 【编程任务】

对于给定的  $n$  个国际机场到中心机场的距离  $d_1, d_2, \dots, d_n$ , 以及地面加油费用  $s$ , 编程计算从距中心机场最远的国际机场飞到中心机场的最少费用。

#### 【要求】

**【数据输入】** 第一行有 2 个整数  $n$  ( $n \leq 400,000$ ) 和  $s$ , 表示有  $n$  个国际机场 (不包括中心机场), 地面加油费用  $s$ 。接下来的 1 行中每行有  $n$  个整数  $d_1, d_2, \dots, d_n$ , 表示给定的  $n$  个国际机场到中心机场的距离。

**【数据输出】** 将编程计算出的最小费用输出。

#### 【样例输入】

5 10

1 3 6 7 10

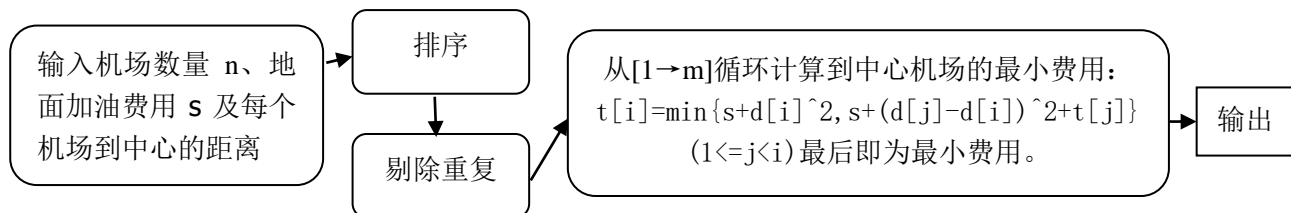
## 【样例输出】

64

## 【算法设计】

(1) **设计思想：** 首先将距离数组 $d[1..n]$ 升序排序，然后将 $d$ 中重复的元素剔除(为了降低时间复杂度)，假设剔除后剩余 $m$ 个元素。另 $t[i]$ 表示机场 $i$ 到中心机场的最小费用，则有递推式： $t[i]=\min\{s+d[i]^2, s+(d[j]-d[i])^2+t[j]\} (1\leq j<i)$ ，从1到 $m$ 递推，最后 $t[m]$ 即最远机场 $m$ 到中心机场的最小费用。

## (2) 设计表示：



## 【调试报告】

刚开始没有剔除重复元素，给程序计算增加了难度，所以之后将重复的元素出去，降低了时间复杂度。

## 【附录】

### 源程序清单：

```
#include "stdafx.h"
#include<iostream>
#include<algorithm>
using namespace std;
int t[400000];
int d[400000];
int main()
{
    int n, s, i, j, len, temp;
    while (cin >> n >> s)
    {
        for (i = 0; i<n; i++)
        {cin >> t[i];}
        sort(t, t + n); //排序
        for (i = 0, j = 1; i<n; i++) //除去重复元素
        {
            if (i == 0)
            {d[j] = t[i];
             j++;}
            else
            {
                if (t[i] != t[i - 1])
```

```

        {d[j] = t[i]; j++;}
    }
}
len = j;
t[1] = s + d[1] * d[1]; //首先计算出第一个机场到中心的最小费用，然后开始递推
for (i = 2; i < len; i++) //t[i]表示机场i到中心机场的最小费用，则有递推式：
t[i] = min{s + d[i]^2, s + (d[j] - d[i])^2 + t[j]} (1 <= j < i)
{
    t[i] = s + d[i] * d[i];
    for (j = 1; j < i; j++)
    {
        temp = s + (d[i] - d[j]) * (d[i] - d[j]) + t[j];
        if (temp < t[i])
            t[i] = temp;
    }
}
cout << t[len - 1] << endl; //输出最小费用
}
return 0;
}

```

测试数据：

5 10

1 3 6 7 10

运行结果：

```

C:\Users\徐鸿飞\Desktop\课程设计\飞机问题\Debug\飞机问题.exe
5 10
1 3 6 7 10
最小费用为: 64

```

## 实习题目八

### 【需求规格说明】

#### 旅游路线

Time Limit: 1000MS Memory Limit: 65536K

Total Submit: 10 Accepted: 7

### 【问题描述】

假如长江沿岸有  $n$  个城市，每个城市依次标号（上游到下游次序编）为  $1, 2, 3, \dots, n-3, n-2, n-1, n$ 。alg 想从长江上游出发，游玩这些城市。其中 alg 的旅游路线选取原则为：



1.至少要游玩一个城市。

2.不会游玩相邻的两个城市。即相邻的两个城市不会出现在 algoo 的旅游路线中。例如：当游玩过城市  $n-k$  后，就不会考虑在城市  $n-k+1$  停下。

现在你的任务是：如果有  $n$  个城市，帮助 algoo 计算有多少种路线可以选择。

### 【要求】

**【数据输入】**多组测试数据。每组测试数据一行，为一个数  $n$  ( $1 \leq n \leq 100$ )，表示城市的个数。

**【数据输出】**对每组测试数据，输出 algoo 总共有多少种路线选择。

### 【样例输入】

3  
4  
5

### 【样例输出】

4  
7  
12  
Hint

数据会好大^^

当  $n=4$  时，有如下几种路线。

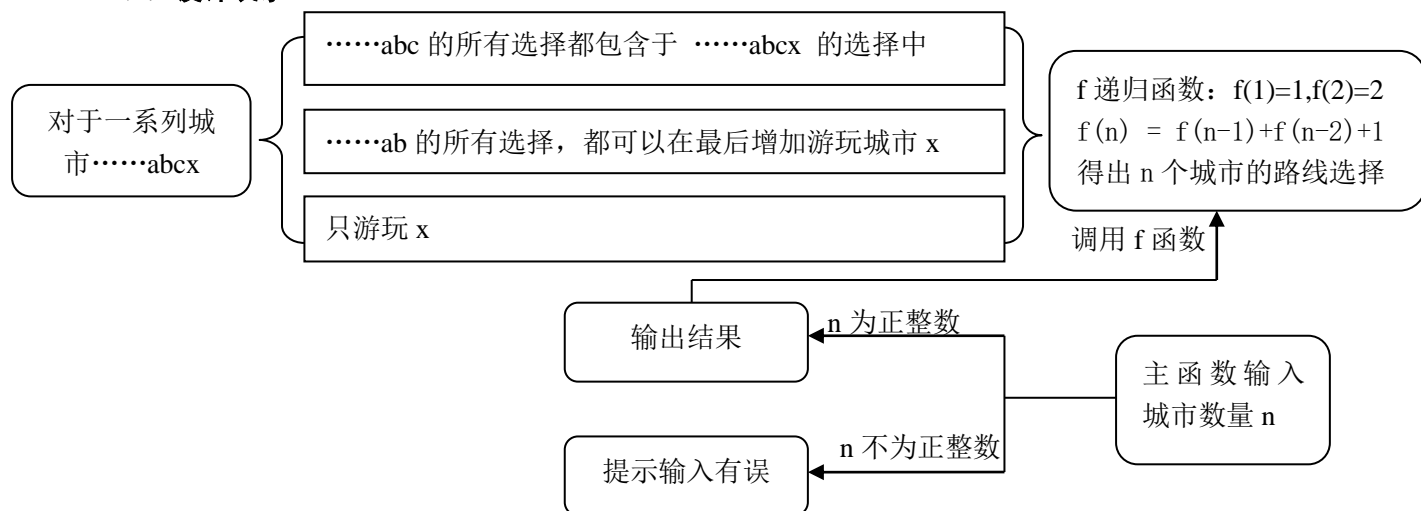
1  
2  
3  
4  
1->3  
1->4  
2->4 (1, 3 城市都不玩，游玩过城市 2 后再到城市 4)  
共 7 种路线。

### 【算法设计】

#### (1) 设计思想：

根据题意：1. 至少要游玩一个城市。 2. 不会游玩相邻的两个城市。 我们可以用递归的思想去做。设在  $n-2$  和  $n-1$  个城市的情况下已经有了结果分别是  $f(n-2)$  和  $f(n-1)$  如已经有  $n-1$  个城市…… $abc$ ，再加入一个城市…… $abcx$ ，则有：经过城市  $c$  的不可以经过城市  $x$ ；没经过城市  $c$  的可以经过城市  $x$ ，也可以不经过城市  $x$ ；以及之前所有城市都没游玩，只游玩了城市  $x$  的。所以当有  $n$  个城市时， $n-1$  个城市的路线的选择包含在  $n$  个城市当中； $n-2$  个城市的选择在选择  $x$ ；以及只选择  $x$  的，所以  $f(n) = f(n-2) + f(n-1) + 1$ 。

## (2) 设计表示:



## 【调试报告】

刚开始没有想到输入错误的情况, 所以当输入的数为非正数时, 程序会崩溃, 所以后来加入了筛选。

## 【附录】

源程序清单:

```
#include "stdafx.h"
#include<iostream>
using namespace std;
int f(int n)//递归函数
{
    if (n == 1)return 1;
    else if (n == 2)return 2;
    else return f(n - 1) + f(n - 2) + 1;
}
int main()//主函数
{
    cout << "请输入城市数量" << endl;
    int n;
    while(cin>>n) //可重复计算多组
    { if (n > 0) //判断是否输入正确的城市数量
    {
        int m = f(n);
        cout << "路线选择有: " << m << "种" << endl;
    }
    else cout << "输入有误" << endl; }
    return 0;
}
```

测试数据：3 4 5

运行结果：

```
C:\Users\徐鸿飞\Desktop\课程设计\旅游路线\Debug\旅游路线.exe
请输入城市数量
3
路线选择有：4种
4
路线选择有：7种
5
路线选择有：12种
```

## 总 结

经过一个学期对《C++语言程序设计》的学习，我学习到了基本的理论知识，了解到了 C++ 语言程序设计的思想，这些知识都为我的课程实践和进一步的学习打下了坚实的基础。本次课程设计，要求完成 8 题，实际完成 8 题，基本上都是控制台程序。个人总体感觉完成情况较好，以下是我对这次课程设计的总结：

- 1、C++语言程序设计是一门更为注重实践的课程，只看书是不行的，虽然我们做了八个学时的课程设计，但我觉得还是远远不够的，我们需要更多的勤加练习，只有更多的练习与实践，我们才能真正的了解 C++ 的思想，更为熟练的运用它去解决实际问题。
- 2、本次实习，对于类的使用不再局限于纸面和书本，渐渐习惯了应用于实际问题。尤其是二级题目，对我加强对类的实际使用起到了很大的作用。
- 3、关于本次课程设计，很多我们所想的和电脑想的是不一样的，过程中因为思路的问题而编译错误的情况很多，有时候这会让人十分沮丧。因此，良好的心态也是我必须要学会的，我们的思想也需要改善，对于语句的使用理解还需要加强，明确电脑“所想”，要知道机器和我们想的不一样，最终实现我们的目的。
- 4、在本次实习中，我明白了代码后面注释的重要性。有时候想要找到自己做的东西却在不清楚的代码中迷失……仅仅百余行的代码就足以让我自己做出来的东西自己找不到，因为编程不仅要自己看懂，也要让他人明白，不仅要编译、敲代码的时候懂，也要在以后打开程序时一目了然。
- 5、另外，我的知识还不够充分，没有学习矩阵，所以所有关于矩阵的题目都只能跳过。

在为期两周的 C++ 课程设计中，我体会颇多，学到了很多。加强了我对 C++ 的认识，并且复习了自己以前学习到的知识。编程是一个持之以恒积累的过程，课程设计是对我第一个学期编程的一个小结，是从理论到实践的一个过程，巩固了我的知识，还学到了一些课本之外的知识，也告诉我，如果不保持经常编程的习惯，就会很快忘掉。真正的程序不是看书看出来的，而是靠代码敲出来的，C++ 是一个终生学习进步的课程，学习并且还需要自己的消化。所以即使课程设计实习结束了，仍然要保持写代码的习惯。

意见和建议：题目的类型可以再广泛一些，有些题目大意模糊不清，课设的老师也可以讲一些例题，以免有的题目根本不知道从哪里下手。