

实验报告

刘思佳，丁玺臻

抄程序

一、细菌实验分组

1、问题分析

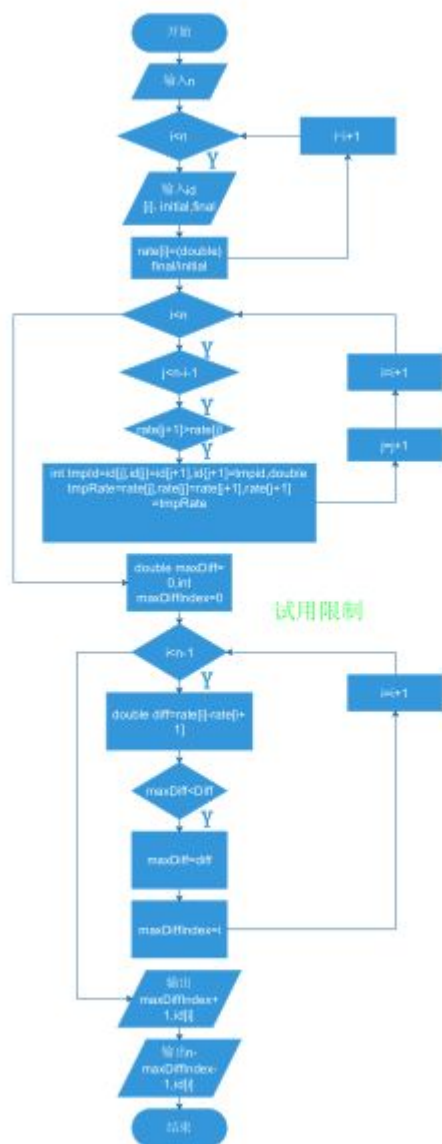
该问题是将外在特征几乎相同、繁殖速度不同的两种细菌，以一个小时为单位求出每个培养皿中细菌的繁殖速率，从而将这两种细菌区分开（依据的原理是：两亚种培养皿之间的繁殖速率有明显的差异，但是同种之间的繁殖速率差别不大）。

2、解决方案

先输入培养皿的个数，然后进行编号，定义初值、末值、繁殖率，然后通过 for 循环和 if 语句按繁殖速率从大到小的顺序将培养皿排序（需要注意的是培养皿序号和繁殖速率都要进行位置变换）；

再通过 for 循环和 if 语句记录下最大的差和最大差的下标，然后将繁殖率大的作为一组并输出组号，将繁殖率小的作为一组也输出组号，这样就实现了对两种细菌培养皿的区分。

3、算法设计



流程图 1

4、编程实现

```

liusijia@liu-pc:~$ vim 细菌实验.cpp
liusijia@liu-pc:~$ g++ -o 细菌实验 细菌实验.cpp
liusijia@liu-pc:~$ ./细菌实验
5
1 10 3456
2 10 5644
3 10 4566
4 20 234
5 20 232
3
繁殖率大1
繁殖率大3
繁殖率大2
2
繁殖率小5
繁殖率小4
liusijia@liu-pc:~$ ./细菌实验
4
1 10 5789
2 10 1234
3 10 254
4 20
456
1
繁殖率大1
3
繁殖率小4
繁殖率小3
繁殖率小2

```

图 1

5、结果分析

对 4 组培养皿进行定义，第一组 rate=578.9、第二组 rate=123.4、第三组 rate=25.4、第四组 rate=22.8，然后按照繁殖率由大到小进行排序，结果是 rate[0]=578.9 rate[1]=123.4 rate[2]=25.4 rate[3]=22.8；

然后再通过 for 循环和 if 语句对繁殖率进行比较，得出 maxDiffIndex=0，随后输出繁殖率较大细菌组和繁殖率较小的细菌组，较大的只有 1 个，即 rate[0]；较小的有 3 个，即 rate[1]、rate[2]、rate[3]，也就是最后的结果，如上图 1。

6、总结体会

相对于之前抄的几个程序来说，这个程序比较复杂，我们在对程序进行理解时要学会进行分段分析，化繁为简、化难为易。

在该程序中尤其要注意 int 和 double 的使用，此外，对细菌的繁殖率进行排序的时候要注意培养皿序号与繁殖率一同转化位置。

编程序

二、苹果和虫子

1、问题分析

该问题是可分为两部分：一、虫子把苹果吃完，则剩余 0 个苹果；二、虫子没有吃完苹果。

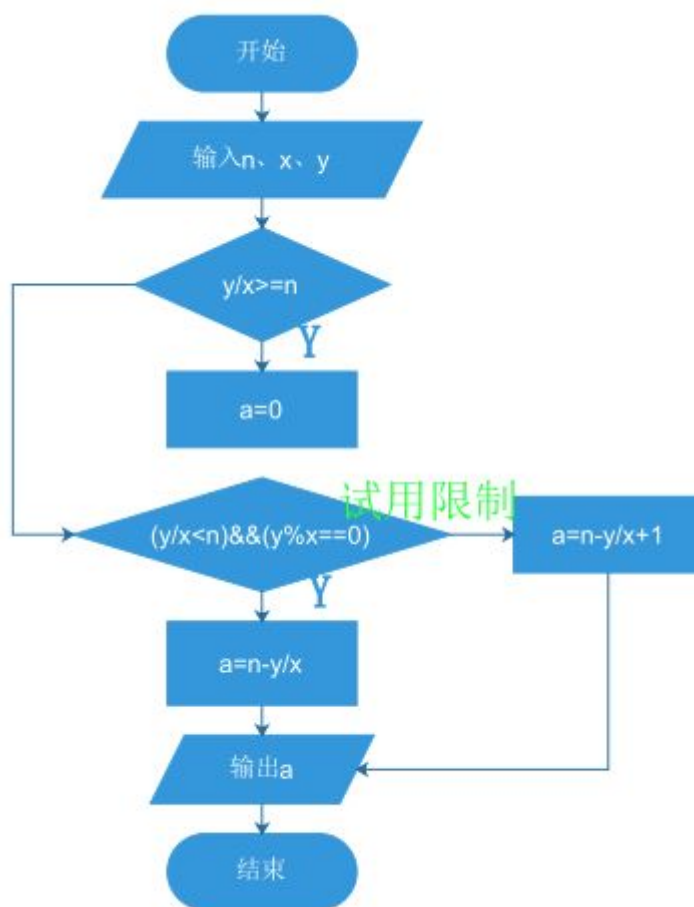
对于第二种情况，又要分为虫子吃最后一个苹果时是否吃的完整。

2、解决方案

(1) 输入苹果总数 n 、虫子吃一个苹果所需时间 x 及虫子一共吃了多长时间的苹果 y ;

(2) 先判定虫子是否吃完苹果, 如果吃完则输出剩余 0 个苹果, 反之进入下一条 if 语句判断虫子吃的最后一个苹果是否完整, 如果刚好吃完最后一个, 则输出 $a=n-y/x$, 反之, 则输出 $a=n-y/x-1$ 。

3、算法设计



流程图 2

4、编程实现

```
#include<iostream>
using namespace std;
int main()
{int n,x,y,a;
cin>>n>>x>>y;
if((y/x)>=n)
a=0;
else
```

```

if((y/x)<n&&(y%x)==0)
a=n-y/x;
else
a=n-y/x-1;
}
cout<<a<<endl;
return 0;
}

```

```

liusijia@liu-pc:~$ vim 苹果与虫子.cpp
liusijia@liu-pc:~$ g++ -o 苹果与虫子 苹果与虫子.cpp
liusijia@liu-pc:~$ ./苹果与虫子
10 2 3
8
liusijia@liu-pc:~$ ./苹果与虫子
10 5 43
1
liusijia@liu-pc:~$ ./苹果与虫子
10 2 30
0
liusijia@liu-pc:~$ █

```

图 2

5、结果分析

第一个结果：一共 10 个苹果，虫子 2 小时吃一个，一共吃了 3 小时，因此剩下的苹果只有 $10 - 3/2 = 8.5$ ，也就是只有 8 个完整的苹果。

第三个结果：一共 10 个苹果，虫子 2 小时吃一个，一共吃了 30 小时，因此剩下的苹果 $10 - 30/2 = -5$ ，也就是没有苹果剩下。

6、总结体会

解决该问题的关键是如何对 if 语句中的条件进行设置，而且该问题中涉及到的都是整数，因此需要 int 来对数据进行整型。

三、大象喝水

1、问题分析

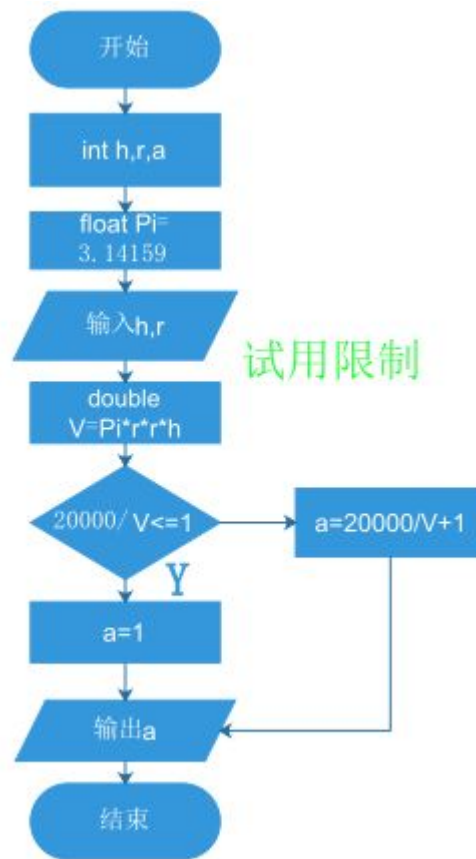
该问题分为两种情况：一、大象一桶就可以喝完；二、大象一桶喝不完，但是由于 Pi 是浮点数，因此最后求得数一定不是整数，需要通过 int 整型，但最后的桶数需要加 1。

2、解决方案

输入桶的底面半径和高，同时定义一个常数 Pi；

设置 if 嵌套语句，首先判断一桶是否能喝完，若能喝完则输出桶数为 1；若不能喝完则输出桶数=大象所喝的水的体积/桶的体积+1。

3、算法设计

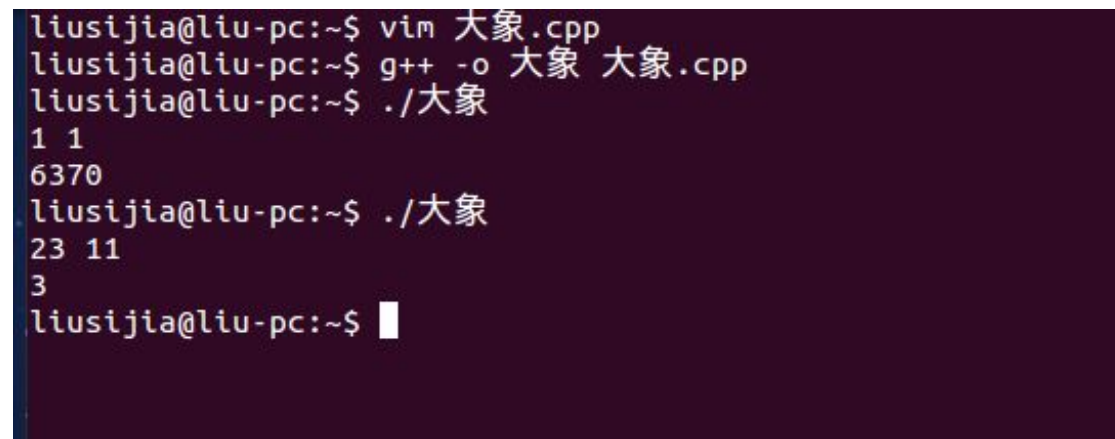


流程图 3

4、编程实现

```
#include<iostream>
using namespace std;
int main()
{
    int h,r,a;
    float Pi=3.14159;
    cin>>h>>r;
    double V;
    V=Pi*r*r*h;
    if((20000/V)<=1)
        a=1;
    else
        if((20000/V)>1)
            a=20000/V+1;
    cout<<a<<endl;
```

```
return 0;
}
```



```
liusijia@liu-pc:~$ vim 大象.cpp
liusijia@liu-pc:~$ g++ -o 大象 大象.cpp
liusijia@liu-pc:~$ ./大象
1 1
6370
liusijia@liu-pc:~$ ./大象
23 11
3
liusijia@liu-pc:~$
```

图 3

5、结果分析

第一个结果：底面半径和高都是 1cm、1cm，因此 $a=20000/(\text{Pi}*1*1*1)+1=6370$ 桶。

第二个结果：高是 23cm、底面半径是 11cm，因此 $a=20000/(\text{Pi}*23*11*11)+1=3$ 桶。

6、总结体会

这道题目要注意题目所给出的单位大象喝的水是 L，桶的底面半径和高是 cm，因此需要进行单位换算；

该题目中最重要的就是 if 语句的嵌套，其次要注意 int 对桶数 a 整型后需要再加一桶。

这个程序虽然简单，但是其与我们的生活有着密切的联系。

四、最高的分数

1、问题分析

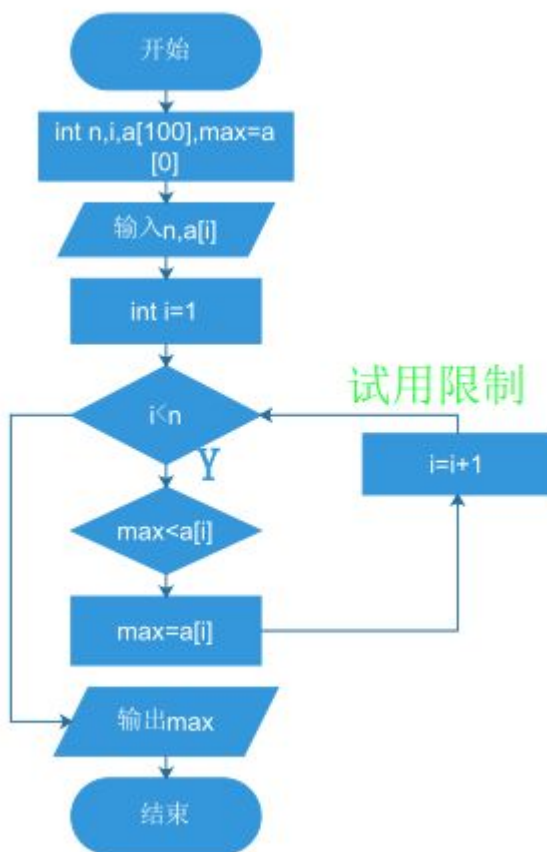
该问题实际上就是求出一个数组中最大的数。

2、解决方案

先通过 for 循环输入需要进行比较的数组；

然后定义一个最大值，假设为 a[0]，然后通过 for 循环和 if 语句实现对整个数组中数据的比较，最后输出最大的那个数

3、算法设计



流程图 4

4、编程实现

```

#include<iostream>
using namespace std;
int main()
{
    int n,i;
    int a[100];
    cin>>n;
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    int max=a[0];
    for(i=1;i<n;i++)
    {
        if(max < a[i])
        {
            max=a[i];
        }
    }
}

```



```

    }
    cout<<max<<endl;
    return 0;
}

```

```

liusijia@liu-pc:~$ vim
liusijia@liu-pc:~$ vim zuida.cpp
liusijia@liu-pc:~$ g++ -o zuida zuida.cpp
liusijia@liu-pc:~$ ./zuida
4
100 55 77 88
100
liusijia@liu-pc:~$ ./zuida
3
12 100 50
100
liusijia@liu-pc:~$ ./zuida
3
99 95 12
99
liusijia@liu-pc:~$

```

图 4

5、结果分析

对 4 个成绩进行比较， $a[0]=100$ 、 $a[1]=55$ 、 $a[2]=77$ 、 $a[3]=88$ 。将 $\text{max}=a[0]=100$ ，然后通过 for 循环、if 语句对这个数组进行比较，最后输出最高分数为 100 分数。

6、总结体会

我们在写该程序的时候遇到了很多问题，刚开始我们使用了冒泡的方法但是没有实现。后来我们用了这个代码：

```

#include<iostream>
using namespace std;
int main()
{
    int n,a[1000],max;
    cin>>n;
    for(int i=0;i<n;i++)
        cin>>a[i]>>" ";
    {
        for(int j=1;j<n;j++)
            if(a[j-1]>a[j])
            {
                max=a[j-1];
                int temp=a[j];
                a[j]=a[j-1];
                a[j-1]=temp;
            }
        else

```

```
max=a[j];  
}  
cout<<max<<endl;  
return 0;  
}
```

这个代码可以在 g++ 中运行，但是在平台中无法通过。

最后我们用了“4、编程实现”中的代码，刚开始未对 max 赋值=a[0]，因此也出现了问题，在历经艰辛后我们终于写出了这个程序。

五、最大奇数与最小偶数之差的绝对值

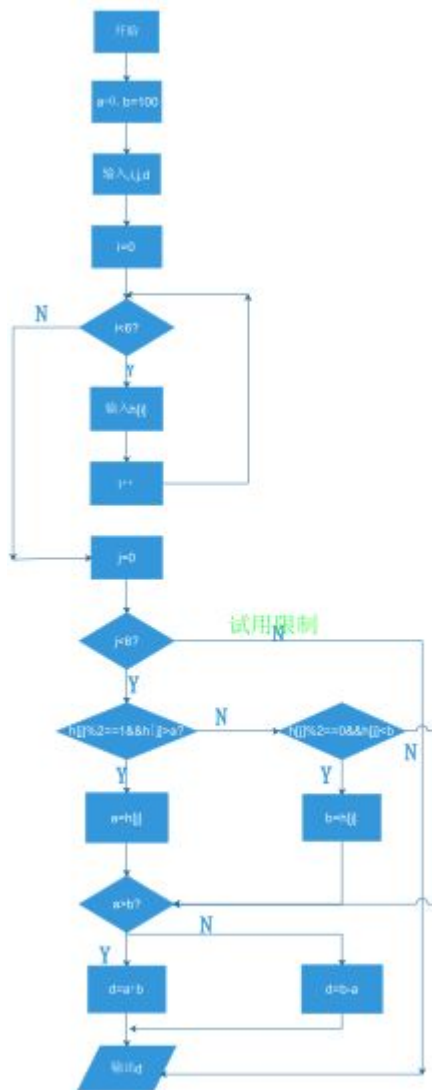
1、问题分析

该问题是求出一个含 6 个数的数组（该数组奇数和偶数的个数都至少为 1）的最大奇数和最小的偶数的差值。

2、解决方案

首先输入这个数组，然后设置 for 循环，并通过 for 循环内部的 if 判断出该数组中最大奇数和最小的偶数。第一个 if 先判断该数是否为奇数且是否大于目前的最大奇数，若不满足该条件则判断是否为偶数且是否小于目前的最小偶数，然后判断最大奇数和最小偶数的大小并求出其差值。

3、算法设计



流程图 5

4、编程实现

```

#include<iostream>
using namespace std;
int main()
{
    int h[5];
    int i,j,d;
    int a=0,b=100;
    for(i=0;i<6;i++)
    {
        cin>>h[i];
    }
    for(j=0;j<6;j++)
    {

```

```

        if((h[j]%2==1)&&h[j]>a)
        {
            a=h[j];
        }
        else
            if((h[j]%2==0)&&h[j]<b)
            {
                b=h[j];
            }
        if(a>b)
            d=a-b;
        else
            d=b-a;
    }
    cout<<d<<endl;
    return 0;
}

```

```

dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ vim 奇偶之差.cpp
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ g++ -o 奇偶之差 奇偶之差.cpp
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ ./奇偶之差
1 2 3 4 5 6
3

```

图 5

5、结果分析

先对该数组的第一个数 1 进行判断，满足第一个 if 语句，因此此时 a=1，依次类推，通过循环对 a、b 不断进行求值得 a=5，b=2，然后求出 a、b 的差值 d=3。

6、总结体会

在编写这个程序的时候要注意不能将 a、b 的初始赋值放到 for 循环中，因为如果这样的话每循环一次都会对 a、b 重新赋值，这样就求不出想要的结果，我们在这方面出了错误。

此外，尤其要注意花括号的使用，因为电脑只会一步步往下执行，不会自动判断你所设置的结构，我们在花括号的匹配上也出了问题，比如 if 和 else 之间的花括号刚开始没有写上就导致程序有问题。

六、分离整数的各个数位

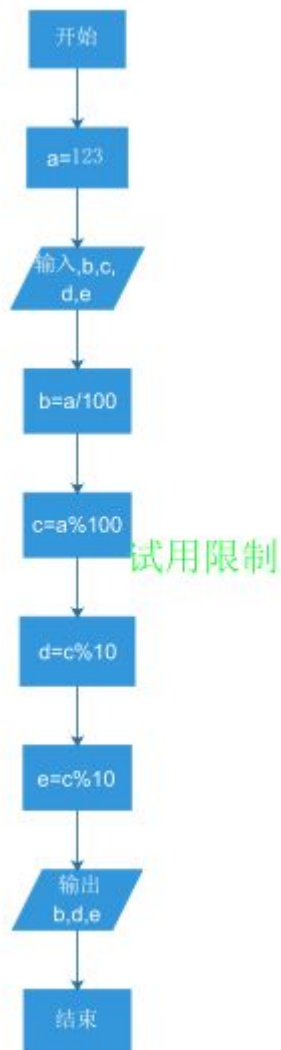
1、问题分析

该问题是求出一个三位数的百、十、个位

2、解决方案

先将该数除 100 并整型，得出该数的百位；
再设置一个中间量，将该数对 100 取余，得出该数的后两位；
然后将该中间量除 10 并整型，得出该数的十位；
最后将中间量对 10 取余，得出个位数。

3、算法设计

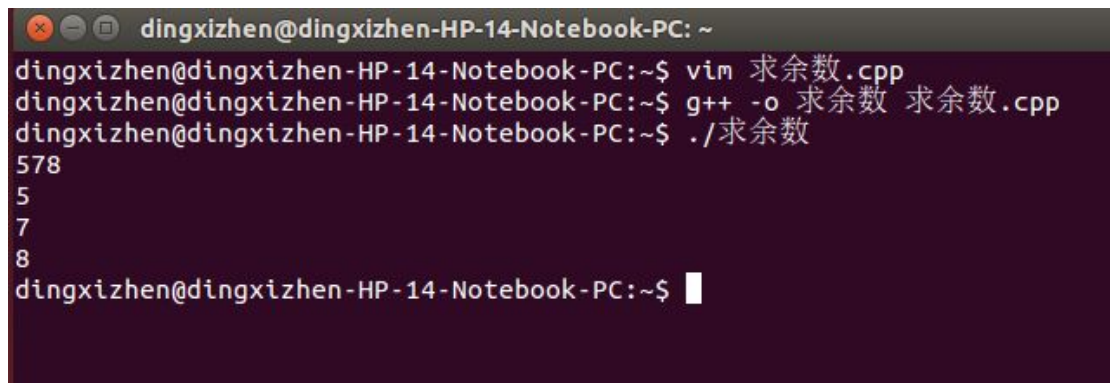


流程图 6

4、编程实现

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c,m,n;
    cin>>n;
```

```
a=n/100;
m=n%100;
b=m/10;
c=m%10;
cout<<a<<endl;
cout<<b<<endl;
cout<<c<<endl;
return 0;
}
```



```
dingxizhen@dingxizhen-HP-14-Notebook-PC: ~
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ vim 求余数.cpp
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ g++ -o 求余数 求余数.cpp
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$ ./求余数
578
5
7
8
dingxizhen@dingxizhen-HP-14-Notebook-PC:~$
```

图 6

5、结果分析

输入一个 3 位数 578，先将 578 除 100 并对结果整型得到其百位数 5，然后将 578 对 100 取余得到其后两位，即 78；再将 78 除 10 得到其十位数 7，最后将 78 对 10 取余，得到其个位数 8。

6、总结体会

该程序总体上来说难度比较低，但是需要分清楚整型、取余运算及除法运算。

体会

在这次作业当中老师开始要求我们去自己编写一些程序，这与上次的抄写程序不同。抄程序，只要求我们把程序看懂，把握住并理解其核心算法和结构就可以，也正如上次的作业题目，感性认识计算机程序，而其目的在于让我们能初步的了解一些编程当中所要用到语句及其功能。然而编写程序却大不相同，在我看来，自己编写程序就好比爬一座山，首先要找到一条可以通向山顶的路，这就好比我们在编程时首先要找到一个合适的算法结构；当然在这过程当中我们会遇到很多的困难，而我们要做的就是坚持下去，直到解决我们遇到的问题。

分工

丁玺臻：分析问题并写出解决方案，进而实现程序。

刘思佳：画流程图并对结果进行分析。

我们二人一起进行总结体会。