

任务六

作者:2014 电子信息工程-范娇娇

2014 电子信息工程-郑春瑞

注明:以下所有程序均已通过 OJ

一、寻找下标

1、问题描述：已知一个元素彼此都不相同的数组 $x[]$,找出给定的数组中是否存在一个元素满足 $x[i]=i$ 的关系，数组下标从零开始。

要求:a、输入：第一行包含一个整数 n ($0 < n < 100$) ,表示数组中元素个数。

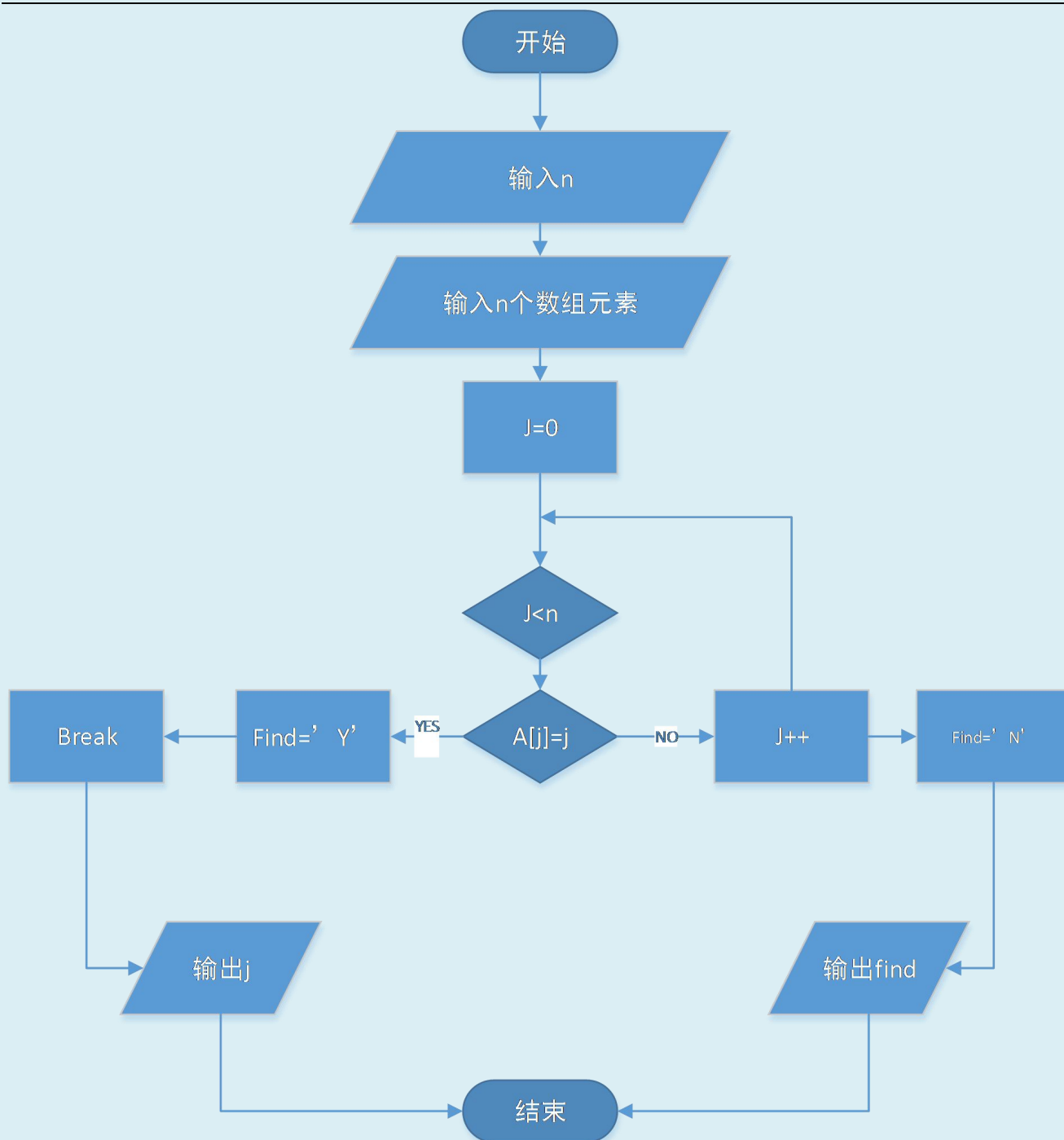
第二行包含 n 个整数，依次表示数组中的元素。

b、输出：输出为一个整数，即满足 $x[i]=i$ 的元素，若有多个元素满足，输出第一个满足的元素。若没有元素满足，则输出 “N”

2、解决方案：(1)首先利用循环输入 n 个数组元素；

(2) 借助一个字符型变量 find 来记录，是否存在满足题意的元素。若存在满足题意的数字元素，记录 find 为 ‘Y’，（在此情况下，输出第一个满足的元素，可在遍历所有元素的情况下，找到第一个元素则跳出此层循环，可利用 break；）若不存在满足题意的元素，则记 find 为 ‘N’；

3.流程图



4.编程实现

```
#include <iostream>
using namespace std;
int main()
{
    int x=0,count1=1,count2=0;
    int n;
    cin>>n;
    int a[100];
```

```
for(int i=0;i<n;i++)
    {cin>>a[i];}
char find = 'N';
for(int j=0;j<n;j++)
{
    if (j== a[j])
    {
        find = 'Y';
        cout << j << endl;
        break;
    }
}
if (find == 'N')
    cout << find<<endl;
return 0;
}
```

样例输入

```
6
-2 -1 7 3 4 8
```

样例输出

```
3
```

5.总结分析：应注意灵活结合使用各种语句，从而达到简化程序的目的。

二、四大湖

1、问题描述:我国有四大淡水湖，一下有四个人的描述：

- A、洞庭湖最大，洪泽湖最小，鄱阳湖第三。
- B、洪泽湖最大，洞庭湖最小，鄱阳湖第二，太湖第三。
- C、洪泽湖最小，洞庭湖第三。
- D、鄱阳湖最大，太湖最小，洪泽湖第二，洞庭湖第三。

这四个湖大小均不相等，四个人每人仅答对一个

请编程实现按照鄱阳湖，洞庭湖，太湖，洪泽湖的顺序给它们的大小排序。

2、解决方案：首先，要保证四大湖的大小不一样，即 a, b, c, d 互不相等；

其次，每个人只说对了一个，可利用逻辑与的运算结果为一来判断。

3、编程实现

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c,d;
    for(a=1; a<=4; a++)
    {
        for(b=1; b<=4; b++)
        {
            for(c=1; c<=4; c++)
            {
                for(d=1; d<=4; d++)
                {
                    if((a!=b&&a!=c&&a!=d&&b!=c&&b!=d&&c!=d)
                        &&((a==3)+(b==1)+(c==2)+(d==4)==1)
                        &&((a==2)+(b==4)+(c==3)+(d==1)==1)
                        &&((b==3)+(d==4)==1)
                        &&((a==1)+(b==3)+(c==4)+(d==2)==1)
                        &&((a+b+c+d)==10))
                    {
                        cout << a << endl << b << endl << c << endl << d<< endl;
                        return 0;
                    }
                }
            }
        }
    }
    return 0;
}
```

样例输出

```
1
2
3
4
```

三、发票统计

1、问题描述：有一个可以统计每个人所报发票总钱数和每类发票总钱数的小型报账系统，假设发票类别共有 A、B、C 三种；一共有三个人，ID 分别为 1、2、3。

要求：a、输入：包括三行，每行第一个数为人员 ID（整形），第二个数为发票总张数（不超过 100），之后是发票类别（字符型）和相应发票金额（单精度浮点型，不超过 1000.0）

b、输出：包含六行，前三行为每人所报发票总钱数（ID 按从小到大的顺序输出，总钱数保留两位小数），后三行为每类发票的总钱数（保留两位小数）

2.解决方案：根据题意，所需要计数的分为每人所报发票总钱数和每种发票总钱数这两类，则遍历输入数据，分类计数。

3.编程实现

```
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    float countA = 0.00;
    float countB = 0.00;
    float countC = 0.00;
    float count1 = 0.00;
    float count2 = 0.00;
    float count3 = 0.00;
    for (int i = 0; i < 3; i++) {
        int ID;
        cin >> ID;
        int n;
        cin >> n;
        for (int j = 0; j < n; j++){
            char kind;
            float money;
            cin >> kind >> money;
```

```
if(kind=='A')
    countA += money;
if (kind == 'B')
    countB += money;
if (kind == 'C')
    countC += money;
if (ID == 1)
    count1 += money;
if (ID == 2)
    count2 += money;
if (ID == 3)
    count3 += money;
}
}
cout << "1 " << fixed << setprecision(2) << count1 << endl;
cout << "2 " << fixed << setprecision(2) << count2 << endl;
cout << "3 " << fixed << setprecision(2) << count3 << endl;
cout << "A " << fixed << setprecision(2) << countA << endl;
cout << "B " << fixed << setprecision(2) << countB << endl;
cout << "C " << fixed << setprecision(2) << countC << endl;
return 0;
}
```

样例输入

```
1 5 A 1.0 A 2.0 C 1.0 B 1.0 C 1
3 3 B 1 C 2 C 1
2 4 B 1 A 1 C 1 A 1
```

样例输出

```
1 6.00
2 4.00
3 4.00
A 5.00
B 3.00
C 6.00
```

四、Tomorrow never knows?

1、问题描述：读入一个格式为 yyyy-mm-dd 的日期，输出这个日期下一天的日期。假定输入的日期不于 1600-01-01，也不晚于 2999-12-30.

要求：输入输出都为一行，格式都为 yyyy-mm-dd，且利用一个字符变量吃掉输入的短横线，输出时请活用 setfill 和 setw 控制符

2.解决方案：一般情况下为当前日期加一即可，以下为特殊情况处理

(1) 闰年 2 月 29 的下一天为 3 月 1 号；平年 2 月 28 号的下一天为 3 月 1 号；

(2) 大月份：1、3、5、7、8、10 月为 31 天，这些月份的 31 号的下一天为下一个月份的一号；12 月 31 号跨年，下一天是下一年的 1 月 1 号

(3) 小月份：4、6、9、11 为 30 天，这些月份的 30 号的下一天为下个月份的 1 号

3.编程实现

```
#include<iostream>
#include <iomanip>
using namespace std;
int main() {
    int year, month, day;
    char a, b;
    cin >> year >> a >> month >> b >> day;
    bool leap = false;
    bool special=false;
    if (year % 4 == 0 & year % 100 != 0)
        leap = true;
    if (year % 400 == 0)
        leap = true;
    if (month == 2) {
        if (leap == true && day == 29) {
            month++;
            day = 1;
            special = true;
        }
        if (leap == false && day == 28) {
            month++;
            day = 1;
            special = true;
        }
    }
}
```

```
if ((month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) && (day == 31))
{
    if (month == 12)
    {
        month = 1;
        day = 1;
        year++;
        special = true;
    }
else {
        day = 1;
        month++;
        special = true;
    }
}
if ((month == 4 || month == 6 || month == 9 || month == 11) && day == 30) {
    day = 1;
    month++;
    special = true;
}
if (special == false)
    day++;
cout << year << '-' << setw(2)<<setfill('0') << month << '-' << setw(2) << setfill('0') << day << endl;
return 0;
}
```

样例输入

2010-07-05

样例输出

2010-07-06

五、流感传染

1、问题描述：有一批易感人群住在网格状的宿舍区内，宿舍区为 $n \times n$ 矩阵，每个格点

为一个房间，可以空着。第一天有些房间的人得了流感，以后每天，得流感的人会使其邻居传染上流感，已经得病的不变，空房间不会传染，请输出第 m 天得流感的人数。

要求：a、输入：第一行输入 n ，不超过 100，表示 $n*n$ 的宿舍房间。接下来的 n 行，每行 n 个字符，‘.’表示第一天该房间住着健康的人，‘#’表示该房间空着，‘@’表示第一天该房间住着得流感的人。接下来的一行是一个不超过 100 的 m 。

B、输出：第 m 天，得流感的人数。

2、解决方案：只有是@的时候，才会对周围产生影响，那么需要对@单独讨论，注意边缘以及上下左右的情况需要考虑清楚。

3、编程实现：

```
#include<iostream>
using namespace std;
int main()
{
    char a[1000][1000];
    int n;
    cin>>n;
    for(int x=0;x<n;x++)
    {
        for(int y=0;y<n;y++)
        {
            cin>>a[x][y];
        }
    }
    int m;
    cin>>m;
    for(int t=0;t<m-1;t++)
    {
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
            {
                if(a[i][j]=='@')
                {
```

```
        if(i-1>=0&& a[i-1][j]=='.')
            a[i-1][j]='*';
        if(j-1>=0&& a[i][j-1]=='.')
            a[i][j-1]='*';
        if(i+1<n&& a[i+1][j]=='.')
            a[i+1][j]='*';
        if(j+1<n&& a[i][j+1]=='.')
            a[i][j+1]='*';
    }
}

for( int r = 0; r< n; r++)
{ for( int k = 0; k < n; k++)
{     if(a[r][k] == '*')
        a[r][k] = '@';}}
}

int s=0;
for(int l=0;l<n;l++)
{
    for(int p=0;p<n;p++)
    {
        if(a[l][p]=='@')
            s++;
    }
}
cout<<s<<endl;
return 0;
}
```

样例输入

```
5
....#
.#.@.
.#@..
#....
.....
4
```

样例输出

```
16
```

六、总结分析

- 1、刚拿到题目的时候，第一反应是比较复杂的，但开始着手解的时候，化繁为简，把一个问题划分为几个部分来分步解决，问题会迎刃而解；
- 2、此次题目正如实验名称，为综合性联系，需要将所有的内容灵活结合使用，从而达到简化程序的目的；
- 3.此次程序多有参考网络程序，在查看网络程序的过程中，学习总结到了一些相应的运用技巧；
- 4、在借鉴的过程中，可以将其和自己的算法相结合，产生新的算法，优化程序；
- 5.此次程序框图由于没有找到合适的表现形式，所以我们没有将这些框图呈现出来。