《C 语言程序设计实验》实验报告

姓　　名： _____

学　　号： _____

班　　级： 统计

成　　绩： _____

一 、题目：

一起去沁湖边赏花吧

Description

武科大有沁湖，与黄家湖相连，是武科大的校内湖，湖中一座长桥连通南北两苑，武科大以沁湖为载体，每年举办沁湖龙舟赛、沁湖诗会、沁湖时光、沁湖讲堂、沁湖鱼宴等校园文化活动。沁湖边上也栽有很多花，小明知道后就一直想去赏花。现在小明得知在沁湖边上有n个点，这n个点从1到n顺时针为围成一圈，每个点栽有一颗花树。有四种花树：梅花（花期1-3月）、樱花（花期3-4月）、石楠花（花期4-5月）、桂花（花期9-10月），每种花只在自己的花期开花。小明给每个点标记一个值p_i，p_i的值表示改点是哪种花，1代表梅花，2代表樱花，3代表石楠花，4代表桂花。小明现在制定了m种观赏方案，每次选定一个月份x，在x月去赏花，在由于沁湖太大，小明就选了湖边两个点u、v，想着就欣赏从u到v沿途的花就好了，因为湖中有桥，现在他发现从u走到v有了几种不同的方式，他想知道沿途最多能看到多少棵树正在开花（走到一个点才能看见这个点的花是否开了，不能重复经过同一个点）。

Input

多组测试数据。

对于每组数据：

输入n，下一行输入n个数$p_1, p_2, \cdots, p_n$。$(2 \leq n \leq 10^5,\ 1 \leq p_i \leq 4)$。

接下来输入两个数p，q表示这两点之间有一座桥$(1 \leq p, q \leq n, p! = q)$。输入$m$，接下来输入$m$行，每行包括$x, u, v$。$(1 \leq m \leq 10^5,\ 1 \leq x \leq 12,\ 1 \leq u, v \leq n,\ u! = v)$

Output

对于m种观赏方案，每次输出能看到的最多的正在开花的树的数量，若数量为零输出`So Sad.`。每次输出占一行。

Sample Input 1

```
10
1 1 2 4 3 2 4 4 3 3
3 9
5
1 1 5
1 3 9
3 5 10
7 1 10
10 2 9
```

Sample Output 1

```
2
2
4
So Sad.
3
```

Source

要求：

（1） 如地点用结构体类型（至少包括种什么类型的花、是否能看到花）

```
struct place
{
    int f_type;//1-4
    int if_open;//0没开，开放
};
```

提示：利用 malloc 建立动态数组

出发点和结束点考虑在桥的同侧还是异侧。

1、实验目的（详细介绍本实验考察的知识点）（30 分）

1.结构体的定义及使用，包括结构体类型，结构体数组，结构体指针，结构体变量的初始化等内容。

2.模块化程序设计，包括子函数的定义及使用，子函数的声明，主函数对子函数的调用，函数调用时的数据传递与参数，子函数返回值的类型，数组作为函数参数，局部变量与全局变量等。

3.数组的定义及其使用，包括数组的初始化，结构体类型数组，字符数组，字符数组的多种专用函数和二维数组的运用等。

4.内存的动态分配，包括了解 malloc,calloc,free 等函数，其中，malloc 可以动态申请内存但不能对其初始化，calloc 包括了初始化的功能更为便利，在程序结束之后用 free 释放指针。

5.循环结构程序设计的运用，熟练掌握 for 循环等循环体结构，了解并运用循环的嵌套，掌握 break 语句等改变循环执行状态的语句，了解何种情况下需要使用循环语句。

6.掌握基本的输入输出函数，了解不同数据类型的区别，掌握算术运算符，关系运算符与关系表达式，逻辑运算符与逻辑表达式，条件运算符与条件表达式等。

7.选择结构的运用，主要为用 if 语句处理实现选择结构，同时也包括用 switch 语句实现多分支选择结构等具体操作要求。

8.函数的嵌套使用。

9.关于指针的运用，包括了解指针是什么，指针变量的引用，通过指针引用数组、字符串等，了解指向函数的指针。

2、代码及其注解（详细写出函数的功能，变量含义，参数含义及返回值类型、语句功能等）（50 分）

```
#include<stdio.h>
#include<stdlib.h>
struct place
{
    int type;
    int open;
}*r;
```

```
//定义花的种类和是否开放的结构体//
int main()
{
    void yue(int m,struct place hua[],int n);
    int begin(int u,int v,int p,int q,struct place hua[],int n);
    int p,q,n,m,a,b,x,u,v,z,y[100][3];
    printf("请输入预计有多少花\n");
    scanf("%d",&n);
    //n 用来存放花的数量及后续动态分配储存空间//
    r=(struct place*)calloc(n,sizeof(struct place));
    //calloc 函数与 malloc 函数比，可以将开辟的动态储存空间初始化//
    printf("请输入预计花的种植分布\n");
    for(a=0;a<n;a++,r++)
    {
        scanf("%d",&(*r).type);
        (*r).open=0;
    }
    //将对应花的种类存放至结构体数组中//
    r=r-n;
    //将指针移回开头方便后续使用//
    printf("请输入预计桥在哪两点，请用逗号隔开\n");
    scanf("%d,%d",&p,&q);
    if(p>q)
    {
        a=p;
        p=q;
        q=a;
```

```
    }
    //确保桥头小于桥尾方便后续运算//
    if(p>n||q>n)
        printf("输入错误\n");
    printf("请输入您制定的方案有多少种\n");
    scanf("%d",&m);
    printf("请依次输入制定的月份，起始点，终止点\n");
    for(a=0;a<m;a++)
    {
        for(b=0;b<3;b++)
            scanf("%d",&y[a][b]);
    }
    //将规划存入 m 行三列的二维数组，第一列月份，第二列起始点，第三列终点//
    for(a=0;a<m;a++)
    {
        yue(y[a][0],r,n);
        //根据输入的月份改变结构体数组中的花的开放状态//
        z=begin(y[a][1],y[a][2],p,q,r,n);
        //二维数组第二列为起始点，第三列为终止点，p,q 为桥头桥尾，r 为结构体指针，n 为结构体中花的数量//
        if(z==0)
            printf("so sad.\n");
        else
            printf("最多的开花的树的数量为%d\n",z);
        //根据子函数传回的数值进行判断//
        for(b=0;b<n;b++,r++)
            (*r).open=0;
        //结束一次循环后让花重新回到都未开放状态避免后续出错//
```

```
        r=r-n;
    }
    free(r);
    system("pause");
    return 0;
}
void yue(int m,struct place hua[],int n)
{
    void meihua(struct place hua[],int n);
    void yinghua(struct place hua[],int n);
    void shinanhua(struct place hua[],int n);
    void guihua(struct place hua[],int n);
    //m 为二维数组第一列的月份//
    switch(m)
    {
    case 1:
    case 2:meihua(hua,n);break;
    case 3:meihua(hua,n),yinghua(hua,n);break;
    case 4:yinghua(hua,n),shinanhua(hua,n);break;
    case 5:shinanhua(hua,n);break;
    case 9:
    case 10:guihua(hua,n);break;
    case 6:
    case 7:
    case 8:
    case 11:
    case 12:break;
```

```
    }
    //根据不同月份传递指针将结构体数组中花是否开放的标记转换//
}
int begin(int u,int v,int p,int q,struct place hua[],int n)
{
    int yixiao(int u,int v,int p,int q,struct place hua[],int n);
    int yida(int u,int v,int p,int q,struct place hua[],int n);
    int qiaoshang(int u,int v,int p,int q,struct place hua[],int n);
    int tong(int u,int v,int p,int q,struct place hua[],int n);
    int i,t,a;
    //u 为起始点，v 为终点，pq 为桥头桥尾//
    if(u>v)
    {
        t=u;
        u=v;
        v=t;
    }
    //确保起点小于终点//
    if(u<p&&v<q)
        i=yixiao(u,v,p,q,hua,n);
    //判断起始点终止点在桥两侧且对应数字小于桥//
    else if(u>p&&v>q)
        i=yida(u,v,p,q,hua,n);
    //判断起始点终止点在桥两侧且对应数字大于桥//
    else if(u==p&&v==q)
        i=qiaoshang(u,v,p,q,hua,n);
    //判断起始点终止点是否和桥对应的点相同//
```

```
    else
        i=tong(u,v,p,q,hua,n);
    //判断起始点终止点是否在桥同侧//
    return(i);
    //根据方案的不同情况转入不同函数，返回值为子函数计算得来的能看到花的最大值//
}
void meihua(struct place hua[],int n)
{
    int i;
    for(i=0;i<n;i++)
        if(hua[i].type==1)
            hua[i].open=1;
    //梅花开放的子函数，无返回值//
}
void yinghua(struct place hua[],int n)
{
    int i;
    for(i=0;i<n;i++)
        if(hua[i].type==2)
            hua[i].open=1;
    //樱花开放的子函数，无返回值//
}
void shinanhua(struct place hua[],int n)
{
    int i;
    for(i=0;i<n;i++)
        if(hua[i].type==3)
```

```
            hua[i].open=1;
    //石楠花开放的子函数，无返回值//
}
void guihua(struct place hua[],int n)
{
    int i;
    for(i=0;i<n;i++)
        if(hua[i].type==4)
            hua[i].open=1;
    //桂花开放的子函数，无返回值//
}
int yixiao(int u,int v,int p,int q,struct place hua[],int n)
{
    //该函数为起始点和终止点分别在桥两侧但是对应的数字比桥对应的数字小的情况//
    int max_1(int a,int b,int c,int d);
    int i,a=0,b=0,c=0,j,x,d=0;
    //u 为起点，v 为终点，pq 为桥头桥尾，n 为花的数量，abcd 对应的不同路线能看到花的数量的计数//
    for(i=u;i<=v;i++)
    {
        if(hua[i].open==1)
            a++;
        if(i==p)
        {
            i=q;
            break;
        }
    }
```

```
    if(i==q)
        for(i;i>=v;i--)
            if(hua[i].open==1)
                a++;
    //顺时针从起点沿着湖走，直到遇到桥时，过桥//
    for(i=u;i<=v;i++)
        if(hua[i].open==1)
            b++;
    //顺时针沿着湖走不过桥到终点//
    for(i=u;i>=0;i--)
        if(hua[i].open==1)
            c++;
    for(j=v;j<n;j++)
        if(hua[j].open==1)
            c++;
    //逆时针沿着湖走不过桥到终点//
    for(i=u;i>=0;i--)
        if(hua[i].open==1)
            d++;
    for(i=n-1;i>=q;i--)
        if(hua[i].open==1)
            d++;
    for(i=p;i<=v;i++)
        if(hua[i].open==1)
            d++;
    //逆时针沿着湖走遇到桥过桥的情况//
    x=max_1(a,b,c,d);
```

```
    return x;
    //返回值为几种路线比较得来的最大值，整型//
}
int yida(int u,int v,int p,int q,struct place hua[],int n)
{
    //起始点和终止点分别在桥的两侧但是对应数字比桥对应的数字大的情况//
    int max_1(int a,int b,int c,int d);
    int i,a=0,b=0,c=0,j,x,d=0;
    for(i=u;i<=v;i++)
    {
        if(hua[i].open==1)
            a++;
        if(i==q)
        {
            i=p;
            break;
        }
    }
    if(i==p)
    {
        for(i;i>=0;i--)
            if(hua[i].open==1)
                a++;
        for(i=n-1;i>=v;i--)
            if(hua[i].open==1)
                a++;
    }
```

```
//顺时针沿着桥走遇桥过桥，过桥后需要逆时针走//
for(i=u;i<=v;i++)
    if(hua[i].open==1)
        b++;
//顺时针沿着湖直接走到终点//
for(i=u;i>=0;i--)
    if(hua[i].open==1)
        c++;
for(j=v;j<n;j++)
    if(hua[j].open==1)
        c++;
//逆时针沿着湖直接走到重点//
for(i=u;i>=0;i--)
{
    if(hua[i].open==1)
        d++;
    if(i==p)
    {
        i=q;
        break;
    }
}
for(i;i<=v;i++)
    if(hua[i].open==1)
        d++;
//逆时针沿着湖遇桥过桥，过桥后变顺时针//
x=max_1(a,b,c,d);
```

```
    return x;
    //返回值为几种路线比较得来的最大值，整型//
}
int qiaoshang(int u,int v,int p,int q,struct place hua[],int n)
{
    //起始点和终止点刚好在桥上的情况//
    int max_1(int a,int b,int c,int d);
    int i,a=0,b=0,c=0,x,j;
    for(i=u;i<=v;i++)
        if(hua[i].open==1)
            a++;
    //顺时针沿着湖走到终点//
    for(i=u;i>=0;i--)
        if(hua[i].open==1)
            b++;
    for(j=v;j<n;j++)
        if(hua[j].open==1)
            b++;
    //逆时针沿着湖走到终点//
    if(hua[p].open==1)
        c++;
    if(hua[q].open==1)
        c++;
    //直接过桥到终点//
    x=max_1(a,b,c,0);
    return x;
    //返回值为几种路线比较得来的最大值，整型//
```

```
}
int tong(int u,int v,int p,int q,struct place hua[],int n)
{
    //起始点和终止点在桥的同一侧的情况//
    int max_1(int a,int b,int c,int d);
    int i,a=0,b=0,c=0,j,x,d=0;
    for(i=u;i<=v;i++)
    {
        if(hua[i].open==1)
            a++;
        if(i==p)
        {
            i=q;
            break;
        }
    }
    if(i==q)
    {
        for(i;i<=v;i++)
            if(hua[i].open==1)
                a++;
    }
    //顺时针沿着湖走遇桥过桥//
    for(i=u;i<=v;i++)
        if(hua[i].open==1)
            b++;
    //顺时针沿着湖走不过桥//
```

```
    for(i=u;i>=0;i--)
        if(hua[i].open==1)
            c++;
    for(j=v;j<n;j++)
        if(hua[j].open==1)
            c++;
    //逆时针沿着湖走不过桥//
    for(i=u;i>=0;i--)
    {
        if(hua[i].open==1)
            d++;
        if(i==p)
        {
            i=q;
            break;
        }
    }
    for(i;i>=v;i--)
        if(hua[i].open==1)
            d++;
    //逆时针沿着湖有桥过桥//
    x=max_1(a,b,c,d);
    return x;
    //返回值为几种路线比较得来的最大值，整型//
}
int max_1(int a,int b,int c,int d)
{
```

```
    if(a<b)
        a=b;
    if(a<c)
        a=c;
    if(a<d)
        a=d;
    return a;
    //将不同路线游赏得到的观赏花的数量进行比较，取最大值//
}
```

3、实验结果及其分析（20 分）

实验结果：依次按照指示输入所要求的数据（花的数量，种类，桥的位置，方案个数，每种方案的出发月份，起始点和终止点等），经多次调试验证，计算机计算所得结果与手动计算结果一致，在依次输入题目所给的数据，结果和题目所给范例相同。

图片参考：

```
请输入预计有多少花
10
请输入预计花的种植分布
1 1 2 4 3 2 4 4 3 3
请输入预计桥在哪两点，请用逗号隔开
3,9
请输入您制定的方案有多少种
5
请依次输入制定的月份，起始点，终止点
1 1 5
1 3 9
3 5 10
7 1 10
10 2 9
最多的开花的树的数量为2
最多的开花的树的数量为2
最多的开花的树的数量为4
so sad.
最多的开花的树的数量为3
请按任意键继续. . .
```

分析：本问题首先要构建起大致框架，即输入指定要求数字后再依次跳转到对应功能的子函数（具体功能详见上面代码），该要求并不难，题目已将要求说得很明白。而在子函数中，则需要依次考虑不同方案对应的不同的分类路线。同时对子函数的处理则需熟练运用结构体指针等内容才能运用子函数将结构体数组中不同花的 open 标签转变为不同数值，以方便后续看到累计开花数量的运算。本问题难点在于方案路线的拆分，需分出多种不同情况，而敲代码时容易计算步骤，分类讨论缺失，忽略一些情况，所以才需要多进行调试，实验多组数据，多种情况，确认运行结果正常，计算结果无误后才可以说代码完成。