**《数据结构》**

**课程设计报告**

|  |  |
| --- | --- |
| 课程名称： | 《数据结构》课程设计 |
| 课程设计题目： | 约瑟夫环 |
| 姓 名： | 应宇杰 |
| 院 系： | 计算机 |
| 专 业： | 计算机科学与技术 |
| 班 级： | 19052312 |
| 学 号： | 19151633 |
| 指导教师： | 葛瑞泉 |

2020年12月1日

1. **需求分析**

#### 功能需求：

1.本演示程序中，集合元素限定为1到100之间，集合大小n<20，集合输入格式以一个回车符为结束标志的字符串。

2.演示程序以用户和计算机的对话方式执行，即在计算机终端显示提示信息之后，用户在键盘上输入演示程序中规定的运算命令相应的输入数据和运算结果显示在其后。

#### 界面需求：

1）请输入人数

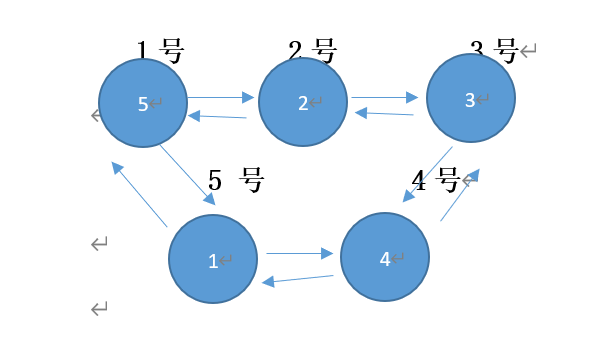
2）输入第一个人的密码直到输入最后一个人的密码

3）输入初始密码。

4）输出结果

1. **概要设计**

这是我定义的一个5人约瑟夫环，用这个例子来模拟约瑟夫环的运行。

****

## 

## 执行5次直到所有数都完成。

Main

基本操作

对结点的操作

creat\_list

Run(运行)

Print（显示）

Createnode(创建结点)

creat\_list（连接结点）

（delte1）删除结点

#### 接口设计

Node\* Creatnode(int i)

操作结果：创建一个双向链表结点,并返回

Node \* creat\_list(int n) //n为双向链表个数

操作结果： 连接n个双向链表，最后形成环，返回一个开头

void delete1(node \*first,node \*last)

操作结果：删除结点，并把删除结点的next和perior连接起来，并输出第i个人的位序；

void loop(int number,node \*head,int n)

操作结果：用while循环输出 ,n是现存的人的个数

并进行了优化，number=number%n;输入的个数用人数取余，如果为0就删除上一个结点。

void delete1(node \*first,node \*last)//输出位序

#### 数据结构设计

ypedef struct node{ //定义一个有前继和后继的双向链表，有位序表和密码

struct node \*next;

struct node \*prior;

int number1;//为位序号

int number2;//为密码

}node;

1. **详细设计**

#include<stdio.h>

#include<stdlib.h>

typedef struct node{

struct node \*next;

struct node \*prior;

int number1;//为位序号

int number2;//为密码

}node;

node\* createnode(int i){//创建结点

node \*p;

int b;

p=(node\*)malloc(sizeof(node));//申请动态内存

if(p==NULL){

printf("动态分配内存失败");

exit(0);

}

printf("请输入第%d个人的密码：",i);

p->number1=i;

scanf("%d",&p->number2);//输入密码

p->next=NULL;

p->prior=NULL;

return p;

}

node\* creat\_list(int n)//连接结点

{

node \* p;

node \* new1;

node \* head;

if(n>=1){

new1=createnode(1);

head=new1;

p=new1;

}

for(int i=2;i<=n;i++){//尾插法插入

new1=createnode(i);

p->next=new1;

new1->prior=p;

p=new1;

}

p->next=head;//形成环

head->prior=p;//形成环

if (n>=1) return head;

}

node\* delete1(node \*p){//删除结点

node \*new2;

int static k=1;//用于标注第几个出列

p->prior->next=p->next;

p->next->prior=p->prior;

printf("第%d个出列的位序是：%d\n",k,p->number1);//输出出列位序

k++;

new2=p->next;

p->next=NULL;

return new2;

}

void loop(int number,node \*head,int n){//勇while循环输出 ,n是现存的人的个数

node \*p=head;

while(p!=NULL)

{

if(number ==0){

p=delete1(p->prior);

number=p->number2;

n--;

}else{

for(int i=2;i<=number;i++) p=p->next;

number=p->number2;

p=delete1(p);

}

}

}

int main(){

int n; //总共有多少人;

int number;

node \*p;

printf("请输入人数： ");

scanf("%d",&n);

p=creat\_list(n);

printf("输入第一个结束值：");

scanf("%d",&number);

loop(number,p,n);

return 0;

}

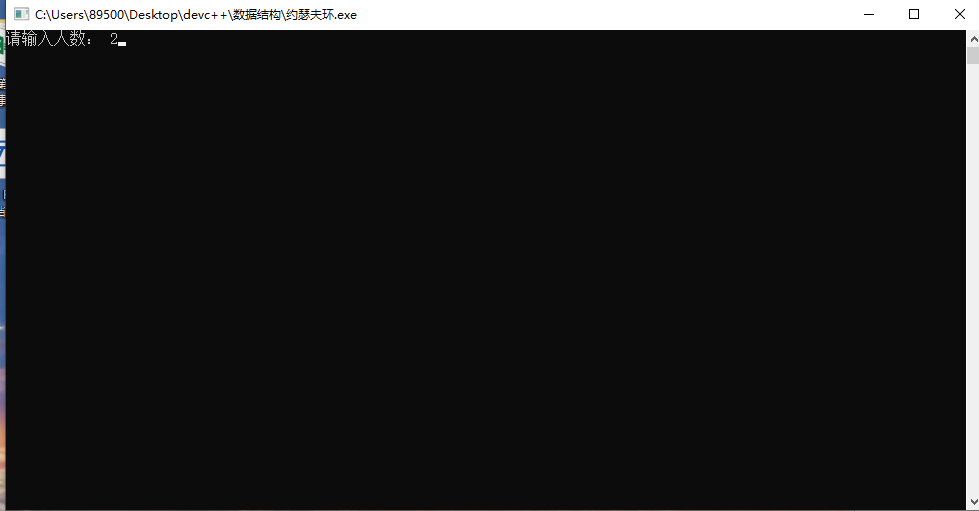
**调试分析**

1.连接时不要忘记成环。

2.测试有没有成环，假如就2 个人，测试第三个人的值和第一个是不是一样就可以。

1. **用户手册**
2. 本程序的执行文件为：约瑟夫环（final）.exe
3. 进入演示程序后，将显示如下的界面

输入人数：



输入第一个人的密码：



输入第二个人的密码：



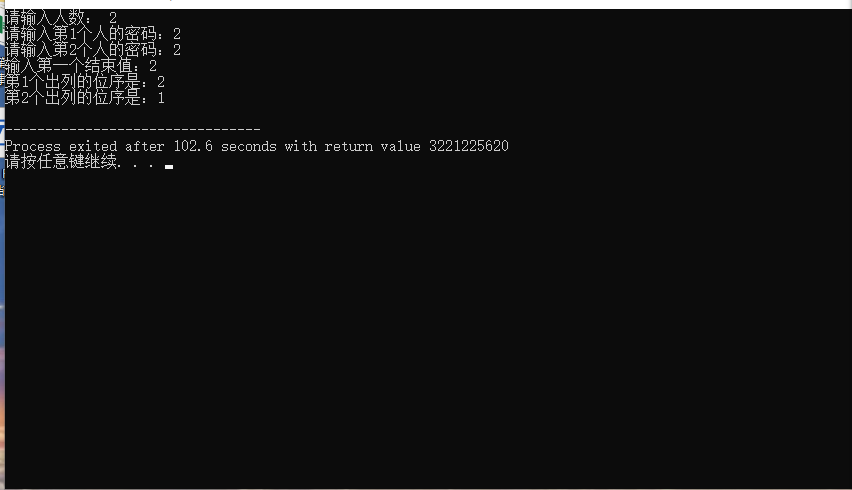
输入第一个结束值：



1. **测试结果**

由于约瑟夫环是打开.exe文件后自动动态显示的。以下取几张截图以作说明

输出第一个和第二个人的位序号



1. **附录**