**《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： | 王一凡 |
| 院 系： | 计算机学院 |
| 专 业： | 软件工程 |
| 班 级： | 19052713 |
| 学 号： | 19061536 |
| 指导教师： | 张灵均 |

2020年11月10日

1. **需求分析**

#### 功能需求：

输入人数的总数，建立双向循环链表，链表中结点的数据域存储生成结点的顺序号和这个人持有的密码。输入初始密码。为了减少程序循环操作，首先对密码进行取余，之后判断遍历链表的方向，满足题目要求时则将此结点从链表中删除，输出它的序号。通过这一系列的循环从而解决约瑟夫环问题。

#### 界面需求：

提示用户输入数据，并且输出每个人出来的顺序。

1. **概要设计**

Createlist（）

判断循环方向

Main

Printflist（）

输出

#### 接口设计

Node Init() //初始化一个双向循环链表

void createlist(Node L,int n) //向链表中存入数据

void printflist(Node L,int t) //循环链表的输出，元素出列

#### 数据结构设计

struct node{

int password,num;//password存储密码，num存储顺序号

struct node \*next,\*prior;

};

typedef struct node \*Node;//双向循环链表

int x; //存储总人数

int m; //存储密码

int sum; //存储此时链表中剩余的结点，即剩余的人数

1. **详细设计**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node{**

**int password,num;**

**struct node \*next,\*prior;**

**};**

**typedef struct node \*Node;**

**//初始化双向循环链表**

**Node Init(){**

**Node head;**

**head = (Node)malloc(sizeof(struct node));**

**head->next=head;**

**head->prior=head;**

**return head;**

**}**

**//向链表中存入数据**

**void createlist(Node L,int n){**

**Node s,r;**

**r=L;**

**int i,x;**

**scanf("%d",&x);//输入第一个密码**

**r->password=x;**

**r->num=1;**

**for(i=2;i<=n;i++){**

**scanf("%d",&x);//输入每个人的密码**

**s=(Node)malloc(sizeof(struct node));**

**s->password=x;//依次输入每个人的密码，并且放到链表中**

**s->num=i;**

**s->prior=r;**

**s->next=r->next;**

**r->next->prior=s;**

**r->next=s;**

**r=s;**

**}**

**}**

**//循环链表的输出，元素出列**

**void printflist(Node L,int t){**

**Node p;**

**p=L;**

**int m,i,k=0,sum;**

**printf("请输入初始密码：");**

**scanf("%d",&m);//m为初始密码**

**printf("出列顺序为：");**

**while(p->next!=p){**

**sum=t-k; //sum为剩余的人数**

**if(m>sum)**

**m=m % sum; //减少循环的次数**

**//判断链表的寻找方向，提高查找的效率**

**if(m<=sum/2){**

**for(i=1;i<m;i++){**

**p=p->next;**

**}**

**printf("%d ",p->num);//输出数据**

**m=p->password;//更新报数值**

**p->prior->next=p->next;//从链表中删除**

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

**L=p->next;**

**free(p);**

**p=L;**

**}else{**

**m=sum-m+1;//从右往左开始查找**

**for(i=1;i<=m;i++){**

**p=p->prior;**

**}**

**printf("%d ",p->num);**

**m=p->password;**

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

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

**L=p->next;**

**free(p);**

**p=L;**

**}**

**k++;**

**}**

**printf("%d",p->num);**

**printf("\n");**

**}**

**int main(){**

**Node L;**

**L=Init();**

**int x,m;**

**printf("请输入人数：\n");**

**scanf("%d",&x);**

**printf("请输入数据：\n");**

**createlist(L,x);**

**printflist(L,x);**

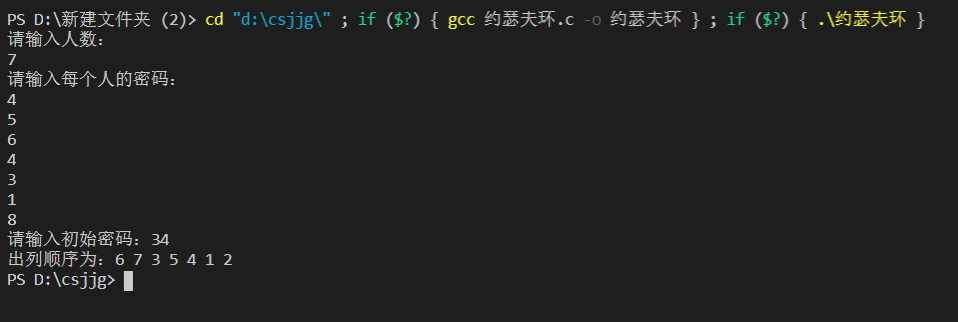
**return 0;**

**}**

1. **调试分析**
2. 本次作业要求求解约瑟夫环问题，题目要求使用双向循环链表的数据结构实现，且需要判断链表的查找方向，极大的提高了查找的效率。
3. 本程序的一个难点在于对查找方向的判断，以及能否想到取余操作，从而减少循环的次数。其次就是对双向循环链表的操作。

3、设置一些提示，告诉用户输入什么数据，并且输出结点的出列顺序。

1. **用户手册**
2. 本程序的执行文件为：约瑟夫环.exe
3. 进入演示程序后，将显示如下的界面，按照程序提示输入数据即可。
4. 
5. **测试结果**



1. **附录**

源程序文件名清单：约瑟夫环.c 约瑟夫环.exe