实习报告

题目:约瑟夫环。

班级：电信9班 姓名：王嘉成 学号：U201813481 日期：2019.3.25

**需求分析。**

1. 本次程序中，用户需输入学生人数、初始上限值、以及各个学生的密码。执行程序后将会按出列顺序输出学生的序号。
2. 执行程序后，计算机会按照约瑟夫环问题的要求，执行步骤，然后输出出列学生的顺序。
3. 测试数据：m=20,n=7,密码=3,1,7,2,4,8,4,输出=6,1,4,7,2,3,5。

m=13,n=5,密码=6,1,4,2,6,输出=3,2,4,1,5。

**概要设计。**

1. 本程序抽象数据类型只有一个线性表。

typedef int Elemtype ;

ADT List{

数据对象:D={ai|ai∈Elemtype,i=1,2,3,…,n,n>=0}

数据关系R1={<ai-1,ai>|ai-1,ai∈D,i=1,2,3,…,n}

基本操作：

Pjos init()

//初始化一个结点

void creat(int n,Pjos L)

//创建单循环链表

其余与本程序无关的基本操作再次就不赘述了。

}ADT List

1. 本程序包括三个模块

1）、主程序模块

Void main()

{

初始化；

Do{

执行约瑟夫环操作；

}while(退出命令)

}

2）、有序表单元模块

3）、节点结构单元模块

各模块调用关系如下：主程序模块→有序表单元模块→节点结构单元模块。

**详细设计。**

以下使是该程序的完整代码：

#include <stdio.h>

#include <stdlib.h>

typedef struct Jos

{

int password;

int num;

struct Jos \*next;

} Jos, \*Pjos;//定义结构体

Pjos init()//创建一个结点

{

Pjos L=(Pjos)malloc(sizeof(Jos));

L->next=NULL;

return L;

}

int password[] = {6,1,4,2,6};

void creat(int n,Pjos L)//创建单循环链表

{

int i;

Pjos head,tail,p;

tail=L;

for(i=0;i<n;i++)//循环创建结点，并将其连接成循环链表

{

p=init();

tail->next=p;

p->password = password[i];

p->num=i+1;

p->next=NULL;

tail=p;

}

tail->next=L->next;//将链表的尾部链接到头部

}

int main()

{

Pjos L;

L=init();

int n,m,i;

int a;

n = 5;

m = 13;

creat(n,L);

Pjos p=L->next, q;

Do//实现约瑟夫环操作

{

for(i=0;i<m-2;i++)//使p指向要删除的结点的上一个结点

{

p=p->next;

}

q=p->next;//删除结点，并输出出列的学生序号

m=q->password;

printf("%d out\n",q->num);

p->next=q->next;

if(m==1){}//注意如果删除的是本身，则p不移向下一个结点

else{p=q->next;}

free(q);

}while(p->next!=p);

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

}

**调试分析。**

1. 创建单循环链表时创建了一个头结点，导致操作时多了一个没有数据域结点，经过检查后发现了此问题，然后给头节点赋值，使它成为满足要求的一个结点。
2. 进行约瑟夫环操作时尝试定位到要删除结点的前面，这样方便删除结点，但当就删除本身时，会出现bug，经过一步一步的顺着程序演算，发现了该错误，然后在操作时添加了一个判断语句，当要删除本身时，就不执行头结点下移的操作。
3. 算法的时空分析：

Creat函数的时间复杂为O(n)，主函数中执行约瑟夫环操作的时间复杂度为O(m\*n)，故总的时间复杂度为O(m\*n)。

**用户使用说明。**

1. 程序的运行环境为Windos操作系统，执行文件为Jos.c。
2. 进入执行页面先输入学生人数，然后输入初始上限值，最后再输入各个学生的密码，之后程序就会输出相应结果。

**测试结果。**

m=20,n=7,密码=3,1,7,2,4,8,4,输出=6,1,4,7,2,3,5。

m=13,n=5,密码=6,1,4,2,6,输出=3,2,4,1,5。

m=5,n=7,密码=8,9,4,2,5,9,8,输出=5,3,1,7,4,2,6。

m=17,n=8,密码=9,0,4,2,9,3,4,6,输出=1,3,7,5,6,4,2,8。

**附录。**

#include <stdio.h>

#include <stdlib.h>

typedef struct Jos

{

int password;

int num;

struct Jos \*next;

} Jos, \*Pjos;//定义结构体

Pjos init()//创建一个结点

{

Pjos L=(Pjos)malloc(sizeof(Jos));

L->next=NULL;

return L;

}

int password[] = {6,1,4,2,6};

void creat(int n,Pjos L)//创建单循环链表

{

int i;

Pjos head,tail,p;

tail=L;

for(i=0;i<n;i++)//循环创建结点，并将其连接成循环链表

{

p=init();

tail->next=p;

p->password = password[i];

p->num=i+1;

p->next=NULL;

tail=p;

}

tail->next=L->next;//将链表的尾部链接到头部

}

int main()

{

Pjos L;

L=init();

int n,m,i;

int a;

n = 5;

m = 13;

creat(n,L);

Pjos p=L->next, q;

Do//实现约瑟夫环操作

{

for(i=0;i<m-2;i++)//使p指向要删除的结点的上一个结点

{

p=p->next;

}

q=p->next;//删除结点，并输出出列的学生序号

m=q->password;

printf("%d out\n",q->num);

p->next=q->next;

if(m==1){}//注意如果删除的是本身，则p不移向下一个结点

else{p=q->next;}

free(q);

}while(p->next!=p);

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

}