**实习报告**

题目：利用单链表编制一个模拟约瑟夫环的程序

班级：电信1809 姓名：朱燚 学号：U201813488 完成日期：2019/3/25

1. 需要分析
2. 本演示程序中，输入值都为int型。
3. 演示程序以用户和计算机的对话方式执行，即在计算机终端上显示“提示信息”之后，由用户在键盘上输入演示程序中规定的运算命令。
4. 程序执行的命令包括：

1）需要输入初始报数上限值。

2）按照顺序输入各人的密码值。

1. 概要设计

为实现上述程序功能，应用循环链表。

1. 循环链表的抽象数据类型定义：

ADT Clinklist{

数据对象：D={

数据关系：R1=

基本操作：

InitList(&L)

操作结果：构造一个空的有序表L。

1. 详细设计

（1）元素类型、结点类型和指针类型

typedef struct node

{

int data;

int code;

struct node \*next;

}node; // 结点类型

node \*h=NULL,\*tail=NULL;

int n,m;

void create(node \*h)//创建链表

{

node \*p;

int i;

for(i=1;i<=n;i++)

{

p=(node\*)malloc(sizeof(node));

p->data=i;

scanf("%d",&p->code);

p->next=NULL;

if(i==1) h=tail=p;

else tail->next=p,tail=p;

}

tail->next=h;

}

（2）主函数和其他函数的伪码算法

int main()

{

//主函数

scanf("%d%d",&n,&m);

create(h);//创建链表

solve(h);//解决问题

return 0;

}//main

void solve(node \*h)

{

node \*p,\*q;

p=tail;

while(p->next!=p)

{

int k=m%n;//转k个人 （取模后）

if(k==0)k=n;

int i;

for(i=1;i<k;i++)

p=p->next;//一个个找到转k个人之后的那个人

q=p->next;// 完成删除前新的关系的建立

m=q->code;

p->next=q->next;

printf("%d ",q->data);

free(q);//删除元素

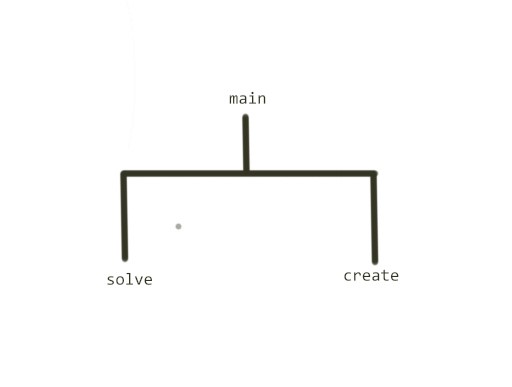
n--;

}

printf("%d",p->data);

}

(3)函数的调用关系图反映了演示程序的层次结构：



1. 调试分析

（1）在定义循环链表的过程中一开始没有使用尾指针，导致构造链表的算法很复杂。

（2）在设计删除结点的算法时利用了头指针，可以保证删去每一个对象。

（3）本程序没有讨论当输入的值并非正整数时候的返回值，需要完善。

（4）算法的时空分析

1）该算法各操作时间复杂度比较合理。

2）基于循环链表实现的算法的各操作的时间复杂度分析如下：

构造算法create读入n个元素，在solve算法中人依次报数，共报k次，k与n数量级相同，故时间复杂度为O（）。

1. 用户手册（略）
2. 测试结果：

执行命令’1’：键入 7 确定人数

执行命令‘2’：键入 20 确定m初值

执行命令‘3’：键入 3 1 7 2 4 8 4 依次输入密码

输出：6 1 4 7 2 3 5 确定出列顺序

1. 附录：

源程序文件名清单（略）