实验一 约瑟夫环

班级:软件工程一班 姓名:秦源 学号:1525161007

1. 需求分析

约瑟夫问题的一种描述为，编号为1,2,3,……n的n个人按顺时针方向围坐一圈，每人持有一个密码（正整数）。一开始任选一个正整数作为报数上限值m,从第一个人开始按顺时针方向自1开始顺序报数，报到m时停止报数。报m的人出列，将他的密码作为新的m的值，从他在顺时针方向上的下一个人开始重新从1报数，如此下去，直至所有的人全部出列为止。试设计一个程序求出出列顺序。

1. 概要设计

利用单向循环链表存储结构模拟此过程，按照出列的顺序印出各人的编号。程序运行后，首先要求用户指定初始报数上限值，然后读取各人的密码。可设n≤30。此题所用的循环链表中不需要“头结点”，请注意空表和非空表的界限。

三、 详细设计

//

// main.cpp

// 约瑟夫环

//

// Created by 秦源 .

// Copyright © 2016年 QinYuan. All rights reserved.

//

#include <iostream>

using namespace std;

unsigned int m;//报数上限值m

unsigned int n;//人数n

struct LNode{

int num;//所持密码

int sq;//人的编码

struct LNode \* next;

};

LNode \* s=new LNode;

void ini(){//初始化

//用于初始化报数上限值(m) 人数(n)

cout<<"请输入初始报数上限值："<<"\t";

cin>>m;

cout<<"请输入人数："<<"\t";

cin>>n;

//获取每个人携带的密码，初始化每个人的编号，并完成单向链表的构建

if(n>0){

cout<<"请输入第"<<1<<"个人所持的密码："<<"\t";

cin>>s->num;

s->sq=1;

//头插法创建链表

if(n>1){

LNode \* tail;

tail=s;

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

LNode \* newOne=new LNode;

tail->next=newOne;

tail=tail->next;

cout<<"请输入第"<<i<<"个人所持的密码："<<"\t";

cin>>tail->num;

tail->sq=i;

}

tail->next=s;

}

}

}

void achieve(){

int number =n;

LNode \* p;

LNode \* tail;

p=s;

cout<<"出列顺序: "<<endl;

while(number>0){

for(int i=1;i<(m-1)%number;i++)

p=p->next;

cout<<"第"<<p->next->sq<<"人"<<"\t";

m=p->next->num;

m++;

tail=p->next;

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

delete tail;

number--;

}

}

int main(int argc, const char \* argv[]) {

ini();

achieve();

cout<<endl;

return 0;

}

四、用户使用说明

测试数据：m的初值为20；n=7,7个人的密码依次为：3,1,7,2,4,8,4,首先m值为6（正确的出列顺序应为6,1,4,7,2,3,5）。

五、 测试结果

