实习报告：题目二

线性表及其应用&约瑟夫环

班级：17计算机一班 姓名：陈新朋 学号：1725111006 完成日期：2018.10.12

1. 需求分析

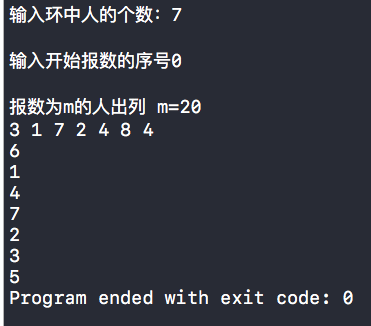
（1）以链表表示线性表。数组中以元素值为int型数据。

（2）用户输入数据，给链表的每个元素赋值。

（3）实现创建链表、删除链表中元素等基本操作。

（4）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。

（5）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）。



（6）程序执行的命令为：

1）创建链表；2）约瑟夫环的操作程序

二、 概要设计

1. 设定线性表的抽象数据类型定义：

ADT list{

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

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

基本操作：

CreateCycList(int n)

操作结果：创建一个单向循环链表，表中节点个数为n。

getout(LinkList head, int n, int m, int k)

初始条件：单向循环链表已存在，表中元素值由键盘随机输入。

操作结果：按约瑟夫环的要求，依次打印出出列的序号。

}ADT list

三、 详细设计

#include <iostream>

using namespace std;

typedef struct Node

{

int data;

int num;

struct Node \*next;

}ListNode,\*LinkList;

LinkList CreateCycList(int n);

void getout(LinkList head,int n,int m,int k);

int main() {

LinkList h;

int n,k,m;

cout<<"输入环中人的个数：";

cin>>n;

cout<<endl<<"输入开始报数的序号";

cin>>k;

cout<<endl<<"报数为m的人出列 m=";

cin>>m;

h=CreateCycList(n);

getout(h, n, m, k);

return 0;

}

LinkList CreateCycList(int n)

{

LinkList head=NULL;

ListNode \*s,\*r;

int i;

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

{

s=new ListNode;

s->data=i;

cin>>s->num;

s->next=NULL;

if(head==NULL)

head=s;

else

r->next=s;

r=s;

}

r->next=head;

return head;

}

void getout(LinkList head, int n, int m, int k)

{

ListNode \*p,\*q;

int i;

p=head;

for(i=0;i<k;i++)

{

q=p;

p=p->next;

}

while(p->next!=p)

{

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

{

q=p;

p=p->next;

}

m=p->num;

q->next=p->next;

cout<<p->data<<endl;

delete p;

p=q->next;

}

cout<<p->data<<endl;

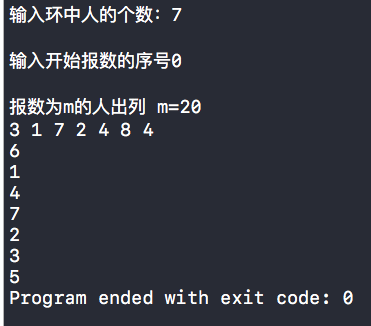
}

四、 调试分析

1．本题中主要算法：getout()函数的时间复杂度为 0（n\*n），

2．经验体会：使用结构体而不是使用类，有时会是编程更加简单。

五、 测试结果



六、 附录

源程序文件名清单：

YSFcircle.cpp//主程序