**实验1 线性表及其应用**

**实验的目的与要求：掌握线性表的定义、不同存储结构及基本运算。**

**实验原理：约瑟夫(Joseph)问题描述为：编号为1，2，3，…，n的n个人按顺时针方向围坐一圈，从第s个人开始从1报数，数到第m的人出列；然后从它在顺时针方向上的下一个人开始重新从1报数，如此下去，直至所有人全部出列为止。设计一个程序求出列顺序。**

**实验步骤：先建立一个循环链表，再按要求实现其功能。**

**参照第（2）个程序完成（1）、（3）的问题代码**

**问题描述：**

**（1）设有n个人围坐成一个圆圈，按一指定方向，从第1个人开始报数，数到m的人出列，然后从下一个开始重新报数，数到m的人又出列，……，如些重复下去，直到n个人全部出列为止。Josephus的问题是：对于任给的n、m和s，求按出列次序得到的n个人员的序列。**

**（2）设有n个人围坐成一个圆圈，按一指定方向，从第s个人开始报数，数到m的人出列，然后从下一个开始重新报数，数到m的人又出列，……，如些重复下去，直到n个人全部出列为止。Josephus的问题是：对于任给的n、m和s，求按出列次序得到的n个人员的序列。**

**参考程序：**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include<iostream.h>**

**typedef struct node //定义结点类型**

**{**

**int num; // 用来对每个结点进行编号标序**

**struct node \*next; //递归定义**

**}LNode,\*Linkls;**

**void create(int m,Linkls &head)**

**{**

**Linkls p;**

**head=(LNode\*)malloc(sizeof(LNode)); //申请头结点的存储空间**

**head->next=head; //让链表成为一个空的循环链表**

**for(int i=m;i>1;i--)**

**{**

**if(!(p=(LNode\*)malloc(sizeof(LNode)))) exit(-1); //申请存储空间**

**p->num=i; //标记每个结点的顺序**

**p->next=head->next;**

**head->next=p;**

**}**

**head->num=1;**

**}**

**int main()**

**{**

**int n,i,s,m,j=1;**

**LNode \*head,\*p,\*q;**

**printf("请输入开始的总人数n：\n");**

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

**printf("开始计数的位置s:\n");**

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

**printf("请输入m的值：\n");**

**scanf("%d",&m);//输入数据**

**create(n,head);//产生一个以head为头结点，有m个元素的循环链表**

**printf("进行依次删除操作：\n");**

**for(i=1,p=head;i<s;i++)**

**p=p->next;**

**while(n--)**

**{**

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

**p=p->next;**

**q=p->next;**

**printf("第%d次出局的数：%d\n",j++,q->num);//输出数到的数字，后面进行删除**

**p->next=q->next;**

**p=p->next;**

**free(q);//释放q所指的空间**

**}**

**return 0;**

**}**

**（3）设有n个人围坐成一个圆圈，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1个人开始报数，数到m的人出列，将他的密码作为新的m值，然后从下一个开始重新报数，数到m的人又出列，……，如些重复下去，直到n个人全部出列为止。Josephus的问题是：对于任给的n、m和s，求按出列次序得到的n个人员的序列。**

**实验要求：**

1. **建立模型模拟约瑟夫环过程，确定存储结构**
2. **输入m,n的值，输出正确序列。**

**要求：参照第（2）个程序完成（1）、（3）的源代码**