实习一 线性表应用

题目： 约瑟夫退圈

班级： 信1901 –4

姓名： 闫竞存

学号： 20194127

电话： 13615475570

日期： 2020 年 10 月 28日

**1. 实验题目**

约瑟夫（Joeph）问题的一种描述是：编号为 1,2,…,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 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，直至所有人全部出列为止。试设计一个程序求出出列顺序。

**2. 需求分析**

任务：利用单向循环链表存储结构模拟约瑟夫退圈过程，按照出列的顺序印出各人的编号。

（1） 程序输入的为不超过int范围的正整数，分别是

（2） 输出的形式

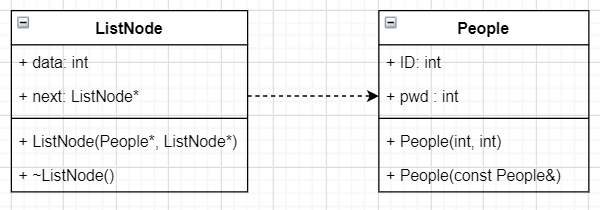
（3）程序所能达到的功能

（4）测试数据

m 的初值为 20；密码：3，1，7，2，4，8，4（正确的结果应为 6，1，4，7，2，3，5）。

**3. 概要设计**

本程序中一共包含了两个类：



**主程序的流程：**

【流程图】

**4. 详细设计**

ListNode::ListNode(People\* d, ListNode\* n) 伪码算法：

ListNode(People\* d, ListNode\* n = NULL) {

data = d;

next = n;

}

ListNode::~ListNode() 伪码算法：

~ListNode() {

if (data) { data = NULL; delete data; }

if (next) { next = NULL; delete next; }

}

People::People(int,int) 伪码算法：

People(int id = 0, int pwd = 0) {

this->ID = id;

this->pwd = pwd;

}

People::People(const Person &) 伪码算法：

People(const People& p) {

this->ID = p.ID;

this->pwd = p.pwd;

}

主程序main() 伪码算法：

**5. 调试分析**

**6. 用户使用说明**

界面输出【指定人数 n (n > 0):】 此时用户输入合法的n表示参与约瑟夫退圈的人数，并敲击回车

界面输出【指定初始上报的上限值 m (m >= 0):】此时用户输入合法的m值表示初始上报的上限值，并敲击回车

界面输出【输入这些人的密码 (密码 > =0)：】此时用户输入n个合法的正整数值，并以空格、tab或者回车分隔，输入完毕之后敲击回车。

此时如果用户操作无误，则界面会输出【出列顺序：】和约瑟夫退圈的顺序，即算法的运行结果。

**7. 测试结果**

测试结果如图所示:

