 实 验 报 告

课程名称： 数据结构（C++版）

题 目： 约瑟夫环问题

专业班级： 18空间

姓 名： 邓珊珊

学 号： 1812001107

指导老师： 袁莹

实验时间： 2020.01.09

第一次实验报告

班级：18空间 姓名：邓珊珊 学号：1812001107

1. 实验目的及原理

（选择公式的原理，及文字说明，所选的公式需要输入什么，计算出什么，如何计算，目的：通过实际问题学会用函数编程实现）

1. 问题描述

设有编号为1，2，···，n的n(n>0)个人围成一个圈，每个人持有一个密码m，从第1个人开始报数，报到m时停止报数，报m的人出圈，再从他的下一个人起重新报数，报到m时停止报数，报m的出圈，······，直到所有人全部出圈为止。当任意给定n和m后，求n个人出圈的次序。

1. 基本要求
2. 建立数据类型，确定存储结构；
3. 对任意n个人，密码为m，实现约瑟夫问题；
4. 出圈的顺序可以依次输出，也可以用一个数组存储。
5. 设计思想

约瑟夫环问题的存储结构。由于约瑟夫环问题本身具有循环性质，考虑采用循环链表，为了统一对表中任意结点的操作，循环链表不带头结点。将循环链表的结点定义为如下结构类型：

struct Node

{

int data; //编号

Node \*next;

};

建立约瑟夫环。建立一个不带头结点的循环链表并由头指针first指示，具体算法与建立但链表相似。

设计约瑟夫环算法实现出圈，伪代码描述：

1. 工作指针pre和p初始化，计数器count初始化；

pre=first->next; count = 2;

1. 循环直到p等于pre
   1. 如果count 等于m,则

2.1.1输出结点p的编号；

2.1.2删除结点p ; p=pre->next；

2.1.3计数器count重新开始计数；

* 1. 否则，执行
     1. 工作指针pre和p后移；
     2. 计数器count增1；

3.退出循环，链表中只剩下一个结点p，输出结点p后将结点p删除。

1. 实验内容

（代码）

头文件 Joseph.h

#ifndef Joseph\_h

#define Joseph\_h

struct Node //定义约瑟夫环的结点Node

{

int data;

struct Node \*next;

};

class JosephRing

{

public:

JosephRing( ); //构造函数，初始化空循环链表

JosephRing(int n); //构造函数，初始化n个结点的环

~JosephRing( ); //析构函数，同单链表析构函数

void Joseph(int m); //密码为m，打印出环的顺序

private:

Node \*rear;

};

#endif /\* Joseph\_h \*/

Joseph.cpp

#include <iostream>

using namespace std;

#include "Joseph.h"

JosephRing :: JosephRing( )

{

rear = nullptr;

}

JosephRing :: JosephRing(int n)

{

Node \*s = nullptr;

rear = new Node;

rear->data = 1; rear->next = rear; //建立长度为1的循环单链表

for (int i = 2; i <= n; i++) //依次插入数据域为2、3、…、n的结点

{

s = new Node; s->data = i;

s->next = rear->next; //将结点s插入尾结点rear的后面

rear->next = s;

rear = s; //指针rear指向当前的尾结点

}

}

JosephRing :: ~JosephRing()

{

if (rear != nullptr){

Node \*p = rear->next;

while(rear->next != rear)

{

rear->next = p->next;

delete p;

p = rear->next;

}

delete rear;

}

}

void JosephRing :: Joseph(int m)

{

Node \*pre = rear, \*p = rear->next; //初始化工作指针p和pre

int count = 1;

cout << "出环的顺序是：";

while (p->next != p) //循环直到循环链表中只剩一个结点

{

if (count < m) { //计数器未累加到密码值

pre = p; p = p->next; //将工作指针pre和p分别后移

count++;

}

else { //计数器已经累加到密码值

cout << p->data<< " "; //输出出环的编号

pre->next = p->next; //将结点p摘链

delete p;

p = pre->next; //工作指针p后移，但pre不动

count = 1; //计数器从1开始重新计数

}

}

cout << p->data << " "; //输出最后一个结点的编号

delete p; //释放最后一个结点

rear = nullptr;

}

Joseph\_main.cpp

#include <iostream>

using namespace std;

#include "Joseph.h"

int main( )

{

int n, m;

cout << "请输入约瑟夫环的长度：";

cin >> n;

cout << "请输入密码：";

cin >> m;

JosephRing R(n);

R.Joseph(m);

return 0;

}

1. 实验成果

（运行结果截图）

