**数据结构上机实习报告**

题目：1.2 约瑟夫环

班级：1403013 学号：14030130061 姓名：刘豪 完成日期：2015.12.04

**一、需求分析**

1． 约瑟夫（Joseph）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，每人可有代表本人的序号。一开始任选一个正整数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，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。

2． 本演示程序中，利用单向循环链表存储结构存储约瑟夫环数据，模拟约瑟夫环的显示过程，按照出列的顺序印出各人的编号。

3． 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（正确的出列顺序为6，1，4，7，2，1，3，5）。

**二、概要设计**

为了实现上述程序功能，我写了一个约瑟夫环的类。

基本操作：

Joseph()

初始化一个空约瑟夫环对象;

Joseph(int n,int m,int tag)

通过模式选择随机生成或者输入一个约瑟夫环;

~Joseph()

析构函数删除对象;

del()

模拟约瑟夫环的节点删除;

print()

打印出环当前的状态，方便调试;

**三、详细设计**

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<ctime>

using namespace std;

class Joseph{

struct Node{

int num,val;

Node \*nex;

};

private:

Node \*head;

int \_n,\_m;

public:

Joseph();

Joseph(int n,int m,int tag);

~Joseph();

void del();

void print();

};

Joseph::Joseph(){

head==NULL;

\_n=\_m=0;

}

Joseph::Joseph(int n,int m,int tag){

int maxn;

if(tag)

scanf("%d",&maxn);

\_n=n;

\_m=m;

head=NULL;

Node \*pre=NULL;

for(int i=1;i<=\_n;i++){

Node \*tmp=new Node;

tmp->nex=NULL;

tmp->num=i;

if(tag)

tmp->val=rand()%maxn+1;

else

scanf("%d",&(tmp->val));

if(head==NULL){

head=pre=tmp;

}

else{

pre->nex=tmp;

pre=pre->nex;

}

}

pre->nex=head;

}

Joseph::~Joseph(){

Node \*tmp;

while(\_n>0){

tmp=head;

head=head->nex;

delete tmp;

\_n--;

}

}

void Joseph::del(){

Node \*tmp=head;

Node \*pre=head;

while(\_n--){

while(--\_m){

pre=tmp;

tmp=tmp->nex;

}

pre->nex=tmp->nex;

printf("%d ",tmp->num);

\_m=tmp->val;

delete tmp;

pre=pre->nex;

tmp=pre;

}

}

void Joseph::print(){

Node \*tmp=head;

printf("n=%d m=%d\n",\_n,\_m);

do{

printf("%d %d\n",tmp->num,tmp->val);

tmp=tmp->nex;

}while(head!=tmp);

}

int main(){

int n,m,tag;

srand((unsigned)time(0)); //设定时间种子

printf("请依次输入人数 (N) 初始位置 (M) :\n");

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

printf("模式1：输入各个节点的密码值:\n");

printf("模式2：输入随机生成密码值的上界:\n");

scanf("%d",&tag);

Joseph test(n,m,tag-1);

test.del();

return 0;

}

**四、调试分析**

1.可选择自由输入和随机生成两种模式进行操作。

2.通过Joseph::print()函数，实时的打印出约瑟夫环的状态，方便调试。

**五、用户手册**

本程序的运行环境为WIN10，执行文件为:约瑟夫环.exe

**六、测试结果**

