

**数据结构课程设计**

**题目：** 约瑟夫生者死者游戏

**专业班级：** 计算机科学与技术 6班

**学 号：** 20204331

**姓 名：** 王博文

**指导教师：** 隋毅

**报告日期：** 2022年3月27日

计算机科学技术学院

**2022年3月27日**

一、课程设计内容

**约瑟夫游戏的大意**：30个游客同乘一条船，因为严重超载，加上风浪大作，危险万分。因此船长告诉乘客，只有将全船一半的旅客投入海中，其余人才能幸免于难。无奈，大家只得同意这种办法，并议定30个人围成一圈，由第一个人数起，依次报数，数到第9人，便把他投入大海中，然后再从他的下一个人数起，数到第9人，再将他投入大海中，如此循环地进行，直到剩下15个游客为止。问：哪些位置是将被扔下大海的位置？

**（1）不失一般性，将30改为一个任意输入的正整数n，而报数上限（原为9）也为一个任选的正整数k。**

**（2）要求使用顺序存储结构和链式存储结构分别实现。**

二、算法设计

在做这次实验之前，我就已经了解到约瑟夫环这个问题，所以我第一时间就想到用求余来控制遍历的下标，本次用到了两种方法，一种是链表做法，一种是数组做法，相比较下，链表做法更加的清楚易懂，但会比数组难写。总的来说这次难度并不是很大。

1. 核心代码实现

#include<iostream>

#include<cstdlib>

using namespace std;

struct Node{

int num;

struct Node \*next;

};

void Joseph(int \*name,int end,int m,int n){

int flag=0;

cout<<"数组方法投入的编号"<<endl;

int i=1,j=1;

while(flag<end){

if(name[j]!=0){

if(i==m){

cout<<name[j];

cout<<" ";

name[j]=0;

flag++;

}

i=i%m+1;

}

j=j%n+1;

}

}

void JosephNode(Node \*head,int end,int m,Node \*tail){

int flag=0;

Node \*q=head->next;

Node \*p=tail;

int i=1;

cout<<"链表方法投入的编号"<<endl;

while(flag<end){

if(i==m){

p->next=q->next;

flag++;

cout<<q->num<<" ";

}

else

p=p->next;

q=q->next;

i=i%m+1;

}

cout<<endl;

}

int main(){

cout<<"依次输入人数与报的数字"<<endl;

int n,m;

cin>>n>>m;

int end=n/2;

int name[n+1];

name[0]=0;

Node \*head;

head=(Node\*)malloc(sizeof(Node));

head->next=NULL;

Node \*p=head;

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

Node \*q;

q=(Node\*)malloc(sizeof(Node));

q->next=NULL;

q->num=i;

p->next=q;

p=q;

name[i]=i;

}

Node \*tail;

tail=p;

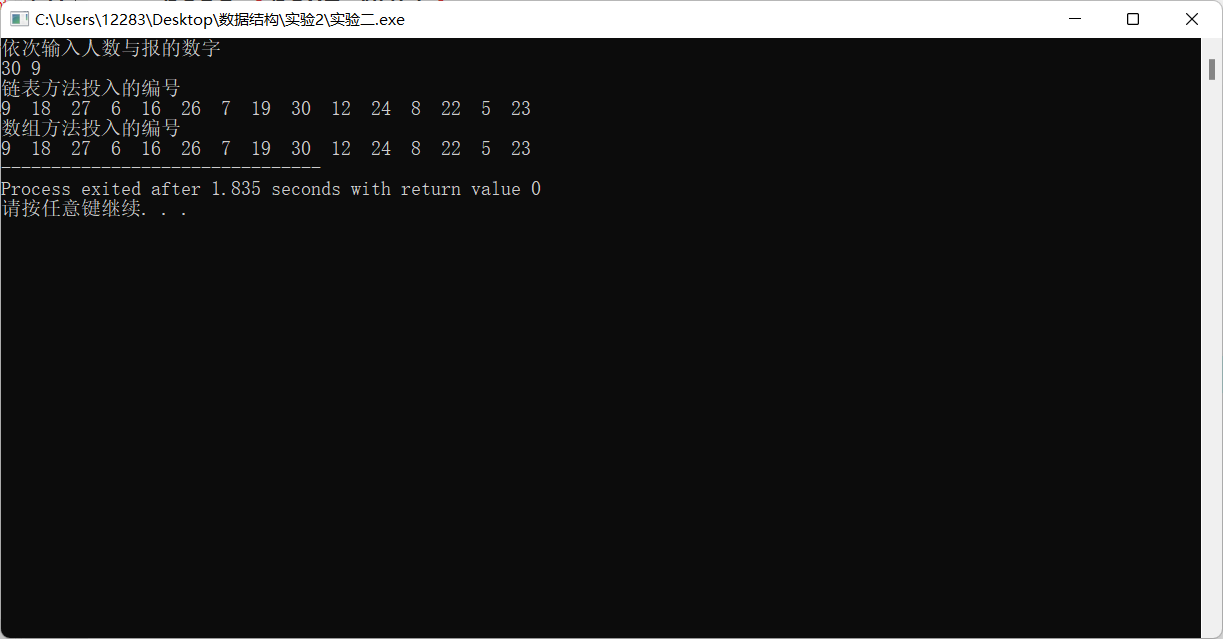
tail->next=head->next;

JosephNode(head,end,m,tail);

Joseph(name,end,m,n);

}

四、测试与分析



五、总结及体会

这次试验虽然不难，但我还是花了不少的时间，主要就是在dev这个个软件的调试方面不是很懂，我最开始的时候是在u盘上写的代码，然后无论怎么调试都不可以，后来才知道需要将本文件放在电脑里才可以，自己的u盘是无法进行调试的

1. 参考书目和网络资源

无