

实 验 报 告

(与程序设计有关)

课程名称: 数据结构

实验题目: 线性表

班级学号: 2203050320

姓 名: 闻家尉

成 绩:

沈 阳 理 工 大 学

2023 年 11 月 1 日

实验目的及要求：

- 1.掌握链表的灵活运用；
- 2.学习链表初始化和建立一个新的单循环链表；
- 3.知道怎样去实现单循环链表删除结点操作；

软硬件环境： 计算机一台，Turbo C 软件或 VC++ 软件

算法或原理分析（实验内容）：

1. 问题描述

编号是 $1, 2, \dots, n (n > 0)$ 的 n 个人按照顺时针方向围坐一圈，每人持有一正整数密码。开始时任选一个正整数作为报数上限值 m ，从某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，从他在顺时针方向的下一个人开始重新从 1 报数，如此下去，直到所有人全部出列为止。令 n 最大值取 30。设计一个程序来求出出列顺序，并输出结果。

2. 基本要求

- ① 建立模型，确定存储结构；
- ② 对任意 n 个人，密码为 m ，实现约瑟夫环问题；
- ③ 出圈的顺序可以依次输出，也可以用数组存储。

3. 算法提示

由于该问题是由古罗马著名史学家 Josephus 提出的问题演变而来，所以通常称之为 Josephus 问题。Josephus 问题的解决需要采用循环链表，先构造一个有 n 个结点的单循环链表，再给出一个报数上限值 m (假设 $m > 1$)，在链表的首结点开始从 1 计数，计到 m 时，对应的结点从链表中删除，然后在被删除结点的下一个结点又从 1 开始计数，直到最后一个结点从链表中删除算法结束。

本设计采用的是不带头结点的循环链表，其中循环链表中结点的结构如下：

```
typedef struct
{
    int num;
    int cipher;
    struct node *next;
}linklist;
```

该问题可由两部分组成，分别由如下两个算法完成：

(1) 建立一个由头指针 **head** 指示的有 n 个结点的约瑟夫单循环链表 **creat**。

(2) 寻找、输出和删除 **head** 所指的单循环链表的第 m 个结点 **select**。该算法由如下具体步骤组成：

- ①在 **head** 中的第一个结点起循环记数找第 m 个结点；
- ②输出该结点的 **num** 值，把该结点的 **cipher**(密码)值赋给 m ；
- ③删除该结点；
- ④转去执行①,直到所有结点被删除为止。

程序代码或实现过程:

```
#include<bits/stdc++.h>
using namespace std;

struct Node
{
    int location;
    int number;
    Node *prior;
    Node *next;
};

struct FuNode
{
    int loc_result;
    int num_result;
    Node *curr;
};

class DoubleLinklist
{
public:
    DoubleLinklist(int count,int number):Count(count),Number(number){}
    void CreateLink();
    void DeleteLink();
    vector<int> node;

private:
    int Count;
    int Number;
    Node *head;
    FuNode Again(Node*);
};

void DoubleLinklist::CreateLink()
{
    Node *head = new Node;
    Node *next;
    Node *curr;
    cout << "请输入第 1 个节点的密码: ";
```

```

cin >> head->number;
head->location = 1;
curr = head;
for(int i=1;i < this->Count;i++)
{
    next = new Node;
    curr->next = next;
    next->location = i+1;
    cout << "请输入第" << i+1 << "个节点的密码: ";
    cin >> next->number;
    next->prior = curr;
    curr = next;
}
next->next = head;
head->prior = next;
this->head = head;
}

FuNode DoubleLinklist::Again(Node *head)
{
    FuNode no;
    int num = 1;
    if(head->next == head)
    {
        no.curr = NULL;
        no.loc_result = head->location;
        no.num_result = head->number;
        cout << "删除的位置为: " << head->location << " 密码为: " << head->number << endl;
    }
    else
    {
        while(num != this->Number)
        {
            head = head->next;
            num++;
        }
        no.curr = head->next;
        no.loc_result = head->location;
        no.num_result = head->number;

        head->prior->next = head->next;
        head->next->prior = head->prior;
        cout << "删除的位置为: " << head->location << " 密码为: " << head->number << endl;
        Number = head->number;
    }
}

```

```
        delete head;
    }
    return no;
}

void DoubleLinklist::DeleteLink()
{
    FuNode no;
    while(this->head != NULL)
    {
        no = Again(this->head);
        this->node.push_back(no.loc_result);
        this->node.push_back(no.num_result);
        this->head = no.curr;
    }
}

int main()
{
    int i,pos;
    cout << "请输入参与游戏的总人数和最开始指定的人的位置： ";
    cin >> i >> pos;
    DoubleLinklist d(i,pos);
    d.CreateLink();
    cout << "出局顺序为： " << endl;
    d.Deletelink();
    return 0;
}
```

结果分析：

```
请输入参与游戏的总人数和最开始指定的人的位置：10 1
请输入第1个节点的密码：9
请输入第2个节点的密码：2
请输入第3个节点的密码：4
请输入第4个节点的密码：6
请输入第5个节点的密码：1
请输入第6个节点的密码：3
请输入第7个节点的密码：8
请输入第8个节点的密码：7
请输入第9个节点的密码：5
请输入第10个节点的密码：10
出局顺序为：
删除的位置为：1 密码为：9
删除的位置为：10 密码为：10
删除的位置为：3 密码为：4
删除的位置为：7 密码为：8
删除的位置为：9 密码为：5
删除的位置为：8 密码为：7
删除的位置为：5 密码为：1
删除的位置为：6 密码为：3
删除的位置为：2 密码为：2
删除的位置为：4 密码为：6
```

教师签字

杨书彤

日期