

实验一 约瑟夫环

PB17111623

范睿

实验要求

1. 假设命令行参数是齐全的且是正确的，运行所编写的程序能正确地输出结果；
2. 能将输出结果导出到文件中。

实验内容

1. 将命令行内传进的参数（字符串格式）转换成整型，赋给各个变量。
2. 根据人数n创建循环单链表L。
3. 模拟报数环节，根据每次停止时的结果在单链表中删除一个结点，将其座位序号输出至文件，并开始下一次报数模拟过程。

关键代码讲述

结点内部结构

```
1 typedef struct CclinkNode{
2     int password; /*密码*/
3     int seat; /*座位序号*/
4     struct CclinkNode *next; /*指向下一结点*/
5 }*cclink;
```

创建单循环链表（无头结点）

```
1 cclink initlist(int n){ /*n参数为总人数，也是要创建的链表表长*/
2     cclink L;
3     struct CclinkNode *p;
4     int i;
5     L = (cclink)malloc(sizeof(struct CclinkNode));
6     p = L;
7     for(i=1;i<n;i++){
8         p->next = (struct CclinkNode*)malloc(sizeof(struct CclinkNode));
9         p = p->next;
10    }
11    p->next = L; /*最后一个结点的后继为第一个结点*/
12    return L;
13 }
```

Joseph函数

此函数的功能为：选出要被删除的结点，将此节点删除，将被删除的结点的座位号输出至文件。

```
1 struct CclinkNode* Joseph(struct CclinkNode* p, int* thresh, FILE* fp){
2     /*p指针指向每一次报数的起点结点; thresh为每次报数的阈值; fp为要写出的文件*/
3     int i;
4     struct CclinkNode* temp = p;
5
6     for(i=0;i<(*thresh)-1;i++){
7         temp = temp->next;
8     }//现在temp指向要被删除的结点
9
10    int seat = temp->seat;
11    *thresh = temp->password;
12
13    if(temp->next == temp) p = NULL;//若链表中只剩下要被删除的结点, 那么删除后p将位NULL
14    else p = temp->next;//否则p指向被删除的结点的下一个节点
15
16    Delete(temp);//将temp结点删除
17
18    printf("%d ", seat);
19    fprintf(fp,"%d ",seat);
20    return p;
21 }
```

Delete函数

```
1 void Delete(struct CclinkNode* p){//p指向要被删除的结点
2     struct CclinkNode* temp = p;
3     while(temp->next!=p)temp = temp->next;//找p的前驱结点
4     temp->next = p->next;//将p从链表中移除
5     free(p);//释放p的空间
6     return;
7 }
```

实验结果及分析

Microsoft Windows [版本 10.0.18362.356]

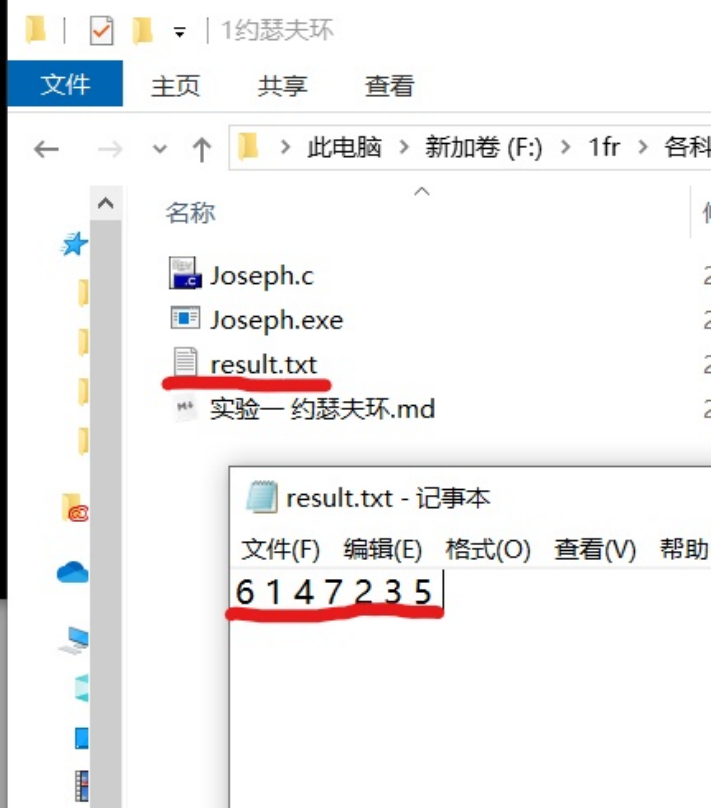
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\fr>F:

F:\>cd F:\1fr\各科作业实验\数据结构\1约瑟夫环

F:\1fr\各科作业实验\数据结构\1约瑟夫环>Joseph.exe 7 20 3 1 7 2 4 8 4
6 1 4 7 2 3 5

F:\1fr\各科作业实验\数据结构\1约瑟夫环>



- 命令提示符中有正确的输出
- 同一文件夹下有result.txt，打开后如右下角所示，也为正确输出

实验小结

在本次试验中，我学会了：

1. 如何用命令行执行可执行文件并正确地传递参数
2. 如何创建单循环链表并进行删除操作

其中感觉自己指针的操作还不太熟练，**argv[]的含义没有把握清楚。经过搜索得到结果：argv可以看做一个数组argv[]，这个数组中每个位置都是一个指向char的指针，如果想要访问指针指向的字符串，可以argv[0], argv[1]...

另外再定义结构体的时候：

```
1 typedef struct cclinkNode{
2
3 }cclinkNode,*cclink;
```

在}的后面写上内容在之后可以直接用来定义变量，比如：

```
1 | CcLinkNode Node;
```

而不用

```
1 | struct CcLinkNode Node;
```

这样。

哈哈。