

# 数据结构项目二文档

同济大学 软件学院 15级2班 1552651 王依睿

## 目录

- [使用说明](#)
  - [操作手册](#)
  - [注意事项](#)
- [概述](#)
  - [项目功能要求](#)
  - [程序设计目的](#)
  - [算法思路](#)
  - [数据结构](#)
  - [文件目录](#)
- [函数接口](#)
  - [循环链表节点 \(Passenger\) 接口](#)
    - [循环链表节点类 \(Passenger\) 代码实现](#)
  - [循环链表 \(PassengerList\) 接口](#)
    - [循环链表类 \(PassengerList\) 代码实现](#)

## 使用说明

### 操作手册

- 运行程序后，先阐述游戏规则。

现有N人围成一圈，从第S个人开始依次报数，报M的人出局，再由下一人开始报数，如此循环，直至剩下K人为止

- 请求用户输入生死游戏的总人数。

```
请输入生死游戏的总人数N: 30
```

- 请求用户输入游戏开始的位置。

```
请输入游戏开始的位置S: 1
```

- 请求用户输入死亡数字。

```
请输入死亡数字M: 9
```

- 请求用户输入剩余生者的人数。

```
请输入剩余的生者人数: 15
```

- 用户将所有信息输入完毕。

```
现有N人围成一圈，从第S个人开始依次报数，报M的人出局，再由下一人开始报数，如此循环，直至剩下K人为止
```

```
请输入生死游戏的总人数N: 30
请输入游戏开始的位置S: 1
请输入死亡数字M: 9
请输入剩余的生者人数: 15
```

- 输出：按死亡顺序显示各死者的位置、最后剩下人数以及剩余死者位置。

```
C:\WINDOWS\system32\cmd.exe
现有N人围成一圈，从第S个人开始依次报数，报M的人出局，再由下一人开始报数，如此循环，直至剩下K人为止

请输入生死游戏的总人数N: 30
请输入游戏开始的位置S: 1
请输入死亡数字M: 9
请输入剩余的生者人数: 15

第0个死者的位置是: 9
第1个死者的位置是: 18
第2个死者的位置是: 27
第3个死者的位置是: 6
第4个死者的位置是: 16
第5个死者的位置是: 26
第6个死者的位置是: 7
第7个死者的位置是: 19
第8个死者的位置是: 30
第9个死者的位置是: 12
第10个死者的位置是: 24
第11个死者的位置是: 8
第12个死者的位置是: 22
第13个死者的位置是: 5
第14个死者的位置是: 23

最后剩下: 15人
剩余生者位置为: 1 2 3 4 10 11 13
14 15 17 20 21 25 28 29
请按任意键继续. . .
```

## 注意事项

- 输入生死游戏的总人数应为正整数。
- 输入游戏开始的位置应为0~生死游戏的总人数间的正整数。
- 输入死亡数字应为0~生死游戏的总人数间的正整数。
- 输入剩余生者的人数应为0~生死游戏的总人数间的正整数。

## 概述

约瑟夫生者死者游戏的大意是：30个旅客同乘一条船，因为严重超载，加上风高浪大危险万分；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，大家只得统一这种方法，并议定30个人围成一圈，由第一个人开始，依次报数，数到第9人，便将他投入大海中，然后从他的下一个人数起，数到第9人，再将他投入大海，如此循环，直到剩下15个乘客为止。问哪些位置是将被扔下大海的位置。

## 项目功能要求

本游戏的数学建模如下：假如N个旅客排成一个环形，依次顺序编号1, 2, ..., N。从某个指定的第S号开始。沿环计数，每数到第M个人就让其出列，且从下一个人开始重新计数，继续进行下去。这个过程一直进行到

剩下K个旅客为止。

本游戏要求用户输入的内容包括：

- 旅客的个数，也就是N的值。
- 离开旅客的间隔书，也就是M的值。
- 所有旅客的序号作为一组数据要求存放在某种数据结构中。

本游戏要求输出的内容是包括：

- 离开旅客的序号。
- 剩余旅客的序号。

## 程序设计目的

实现一个遍历循环链表并删除指定节点的简单程序，练习循环链表的编写实现，熟悉面向对象程序设计。

## 算法思路

以循环遍历为基本算法，而后通过改变节点的指针实现删除。

## 数据结构

- 将循环链表节点封装成一个类
- 将循环链表封装成一个类

## 文件目录

- 2\_1552651\_wangyirui.cpp（主文件）
- passenger\_class.h（循环链表节点类）
- passengerlist\_class.h（循环链表类）
- 2\_1552651\_wangyirui.exe（可执行文件）
- 2\_1552651\_wangyirui.pdf（项目文档）

---

## 函数接口

## 循环链表节点（Passenger）接口

成员函数名	公有/私有性	功能	参数	返回值类型
Passenger(int num)	public	构造函数，初始化类内成员变量 num_	序号 int	/
int num()	public	返回类内成员变量num_	/	int

## 循环链表节点类（Passenger）代码实现

```
//linked list's node
class Passenger {

public:
    Passenger(int num) { num_ = num; }
    int num() { return num_; }

public:
    Passenger *next_ = NULL;

private:
    int num_;
};
```

## 循环链表（PassengerList）接口

成员函数名	公有/ 私有 性	功能	参数	返回 值 类 型
PassengerList(int total, int startnum, int deathnum, int residue)	public	构造函数，初始化类内成员变量，用new方法为链表的头指针开辟空间	总人数int，游戏开始的位置int，死亡数字int，剩余生者的人数int	/
~PassengerList()	public	析构函数，用delete方法释放前面用new方法开辟的空间	/	/
int residue()	public	返回类内成员变量residue_	/	int
void Create()	public	利用循环，用new方法为新节点开辟空间，用所有节点构造一个循环链表	/	/
void Kill()	public	循环遍历循环链表，根据死亡数字依次选出死者，打印死者的序号，将死者节点从循环链表中删除，并用delete方法释放为此节点开辟的空间	/	/
void SurvivalListOutput()	public	打印最后剩下的生者人数，依次打印生者的位置	/	/

## 循环链表类（PassengerList）代码实现

```

//The linked list and operations
class PassengerList {

public:
    //initialize with arguments offered
    PassengerList(int total, int startnum, int deathnum, int residue) :
        total_(total),startnum_(startnum),deathnum_(deathnum),residue_(residue){
        head_ = new Passenger(total_);
    }
    //delete the nodes
    ~PassengerList() {
        auto current = head_->next_;
        Passenger *following = NULL;

        while (current != head_) {
            following = current->next_;
            delete current;
            current = following;
        }
        delete current;
    }

public:

    int residue() { return residue_; }

    //create the linked list
    void Create();
    //kill the unlucky guys
    void Kill();
    //print the lucky guys
    void SurvivalListOutput();

private:
    //used as the tail of the linked list
    Passenger *head_;
    int total_;
    int startnum_;
    int deathnum_;
    int residue_;
};

```