

# 约瑟夫生者死者游戏

1452764 何冬怡

## 目录

项目要求 .....	2
使用手册 .....	3
程序概述 .....	3
1. 数据结构 .....	3
2. 算法思路 .....	3
3. 文件目录 .....	3
成员变量/函数接口 .....	4

## 项目要求

采用单循环链表。

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  $1, 2, \dots, N$ 。从某个指定的第  $S$  号开始。沿环计数，每数到第  $M$  个人出列，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：**

- 1、旅客的个数，也就是  $N$  的值；
- 2、离开旅客的间隔书，也就是  $M$  的值；
- 3、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包括：**

1. 离开旅客的序号；
2. 剩余旅客的序号。

## 使用手册

1. 打开 exe 文件后，按提示输入；
2. 得到输出结果。

```
请输入生死游戏的总人数: 30
请输入游戏开始的位置: 1
请输入死亡数字M: 9
请输入剩余的生者人数K: 15
第1个死者的位置是: 9
第2个死者的位置是: 18
第3个死者的位置是: 27
第4个死者的位置是: 6
第5个死者的位置是: 16
第6个死者的位置是: 26
第7个死者的位置是: 7
第8个死者的位置是: 19
第9个死者的位置是: 30
第10个死者的位置是: 12
第11个死者的位置是: 24
第12个死者的位置是: 8
第13个死者的位置是: 22
第14个死者的位置是: 5
第15个死者的位置是: 23

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

3. 注意：本程序利用模拟实现，对于大数据量可能运算较慢。

## 程序概述

### 1. 数据结构

将每个游戏者的位置、生死和后继指针作为节点类 Player 封装；由于数据量确定，故使用动态数组储存所有的节点类以便输出生还者信息。单向循环链表类封装为 PlayerList。

### 2. 算法思路

单向循环链表模拟游戏过程，死者移除出链表。数组顺序遍历输出生者。

### 3. 文件目录

可执行文件	2_1452764_hedongyi.exe
主文件	2_1452764_hedongyi.cpp
类定义声明	2_1452764_hedongyi.h
项目文档	2_1452764_hedongyi.pdf

## 成员变量/函数接口

成员变量名	数据类型	功能说明	
_pos	Int	位置	
_alive	bool	是否存活（true 表活着），初始化为 true	
_next	Player*	后继指针	
成员函数名	功能	参数	返回值
Player()	默认构造函数		
Player(int)	重载的带参构造函数，生成对应位置玩家	int 位置	
next()	获得后继指针	空	Player*
setNext(Player*)	修改后继指针	Player* 目标后继指针	void
die()	玩家死亡，置_alive 为 false	空	void
setpos(int)	修改位置	int 目标位置	void
getpos()	读取位置	空	int
isAlive()	读取是否存活	空	bool

表 1 Player 类接口

成员变量名	数据类型	功能说明	
head	Player*	循环链表头指针；数组头指针	
number	int	总人数	
成员函数名	功能	参数	返回值
PlayerList()	默认构造函数		
PlayerList(int)	构造一定数量节点的链表	int 节点数量	
startGame(int start, int gaps, int k)	开始模拟游戏	int start 开始位置 int gaps 间隔人数 int k 存活人数	void
printAlive()	标准输出存活的位置	空	void

表 2 EDULIST 类接口