**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机系统1**  **项目名称 实验四和五 Nim游戏**  **学 院 计算机与软件学院**  **专 业 软件工程**  **指导教师 陈飞**  **报 告 人 学号**  **实验时间 2025年5月29日**  **提交时间 2025年6月3日** |

**教务处制**

# 一、实验目的与要求

**实验要求**

⑴ 在游戏开始时，你应该显示游戏界面的初始化状态。具体包括：在每行石头的前面，你应该先输出行的名称，例如“ROW A”。你应该使用ASCII字符小写字母“o”（ASCII码 x006F）来表示石头。游戏界面的初始化状态应该如下：

*ROW A: ooo*

*ROW B: ooooo*

*ROW C: oooooooo*

⑵ 游戏总是从玩家1先开始，之后玩家1和玩家2轮流进行。在每一个回合开始时，你应该输出轮到哪一个玩家开始，并提示玩家进行操作。例如，对于玩家1，应该有如下显示：

*Player 1,choose a row and number of rocks:*

⑶ 为了指定要移除哪一行中的多少石头，玩家应该输入一个字母后跟一个数字（输入结束后不需要按Enter键），其中字母（A,B或C）指定行，数字（从1到所选行中石头的数量）指定要移除的石头的数量。你的程序必须要确保玩家从有效的行中移除有效数量的石头，如果玩家输入无效，你应该输出错误提示信息并提示该玩家再次进行输入。例如，如果轮到玩家1：

*Player 1, choose a row and number of rocks: D4*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: A9*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: A\**

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks: &4*

*Invalid move. Try again.*

*Player 1, choose a row and number of rocks:*

你的程序应保持提示玩家，直到玩家选择有效的输入为止。确保你的程序能够回显玩家的输入到屏幕上，当回显玩家的输入后，此时应该输出一个换行符（ASCII码x000A）使光标指向下一行。

⑷ 玩家选择有效的输入后，你应该检查获胜者。如果有一个玩家获胜，你应该显示相应的输出来表明该玩家获胜。如果没有胜利者，你的程序应该更新游戏界面中每行石头的数量，重新显示更新的游戏界面，并轮到下一个玩家继续。

⑸ 当某个玩家从游戏界面上移除最后的石头时，游戏结束。此时，你的程序应该显示获胜者然后停止。例如，如果玩家2移除了最后的石头，你的程序应该输出一下内容：

*Player 2 Wins.*

**样例输入/输出**

**注意：你的程序中输入输出的格式必须完全和样例中的格式相一致**

ROW A: ooo

ROW B: ooooo

ROW C: oooooooo

Player 1, choose a row and number of rocks: B2

ROW A: ooo

ROW B: ooo

ROW C: oooooooo

Player 2, choose a row and number of rocks: A1

ROW A: oo

ROW B: ooo

ROW C: oooooooo

Player 1, choose a row and number of rocks: C6

ROW A: oo

ROW B: ooo

ROW C: oo

Player 2, choose a row and number of rocks: G1

Invalid move. Try again.

Player 2, choose a row and number of rocks: B3

ROW A: oo

ROW B:

ROW C: oo

Player 1, choose a row and number of rocks: A3

Invalid move. Try again.

Player 1, choose a row and number of rocks: C2

ROW A: oo

ROW B:

ROW C:

Player 2, choose a row and number of rocks: A1

ROW A: o

ROW B:

ROW C:

Player 1, choose a row and number of rocks: A\*

Invalid move. Try again.

Player 1, choose a row and number of rocks: &4

Invalid move. Try again.

Player 1, choose a row and number of rocks: A1

Player 1 Wins.

----- Halting the processor -----

**提示与建议**

⑴ 记住，程序中所有的输入输出使用ASCII字符，你应该负责进行必要的转换。

⑵ 从键盘中输入字符你应该使用TRAP x20（GETC）指令，同时为了回显输入的字符到屏幕上，你应该使用TRAP x21（OUT）指令，该指令紧跟在TRAP x20指令之后。

⑶ 你应该在适当的时候使用子程序。

⑷ 在你编写的每个子程序中，应该保存并还原所使用的任何寄存器。这将避免你在调试过程中遇到问题。

⑸ 在一个回合中，玩家的输入必须包含指定为A,B或C（即大写字母）的行，后面紧跟不大于该行仍然存在的石头数量的数字。

**提示：**

① 你应该设置程序的开始地址在x3000（如,程序的第一行指令应该为 .ORIG x3000）

② 源文件命名为**nim.asm**

# 二、实验内容与方法

1.主程序实现游戏的基本交互逻辑，包括盘面输出、玩家输入处理、胜负判断及结果显示，通过操作寄存器和子程序调用完成游戏流程控制。

2.界面展示子程序DISPLAY输出游戏盘面的行提示（ROW A/B/C）及对应行的石头数，模块化展示游戏盘面，为玩家提供清晰的界面反馈，与输入处理和胜负判定配合，构成完整的游戏交互逻辑。

3.球数展示子程序PUTBALL与界面展示子程序配合，循环输出球，输出游戏盘面。

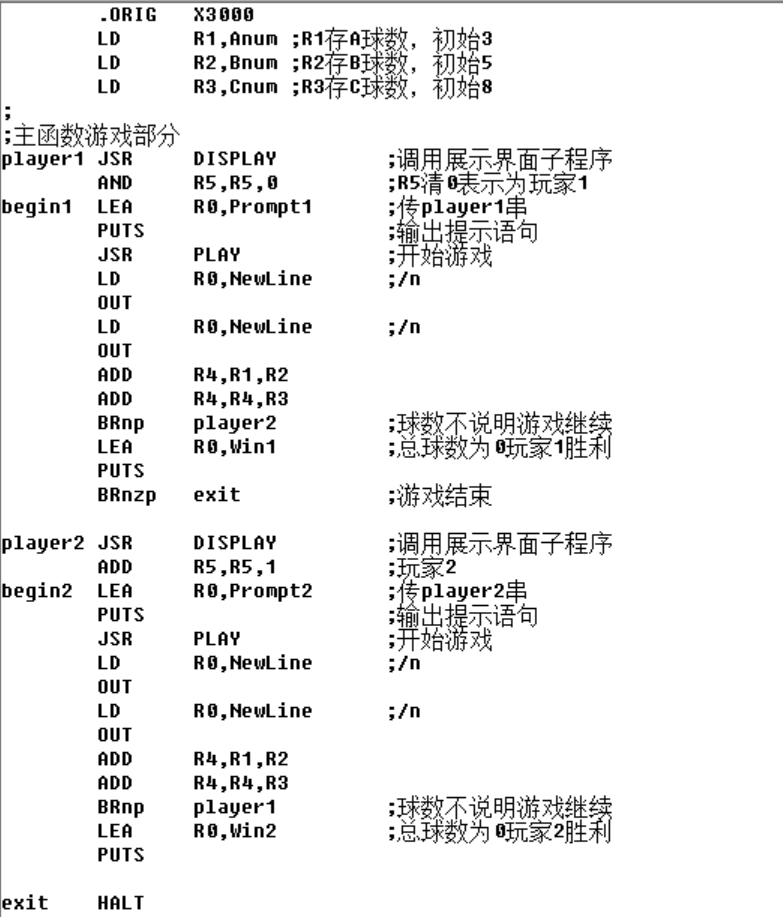
4.单人每轮子程序PLAY根据输入的行名和球数判断是否有效，实现单轮交互，验证输入合法性，更新盘面，处理错误。

# 三、实验步骤与过程

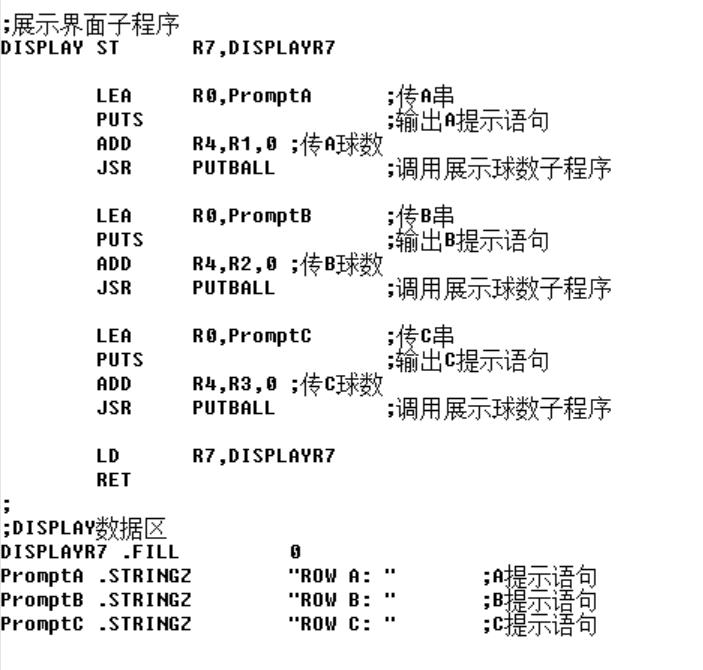
（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

一）程序实现

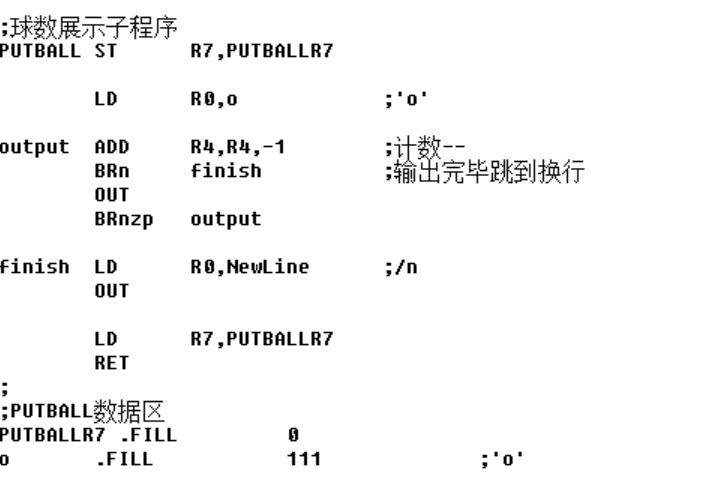
1.主程序，调用子程序，实现游戏的核心逻辑：玩家轮流取球，更新盘面后检查总球数，为 0 时判定当前回合的玩家胜利。



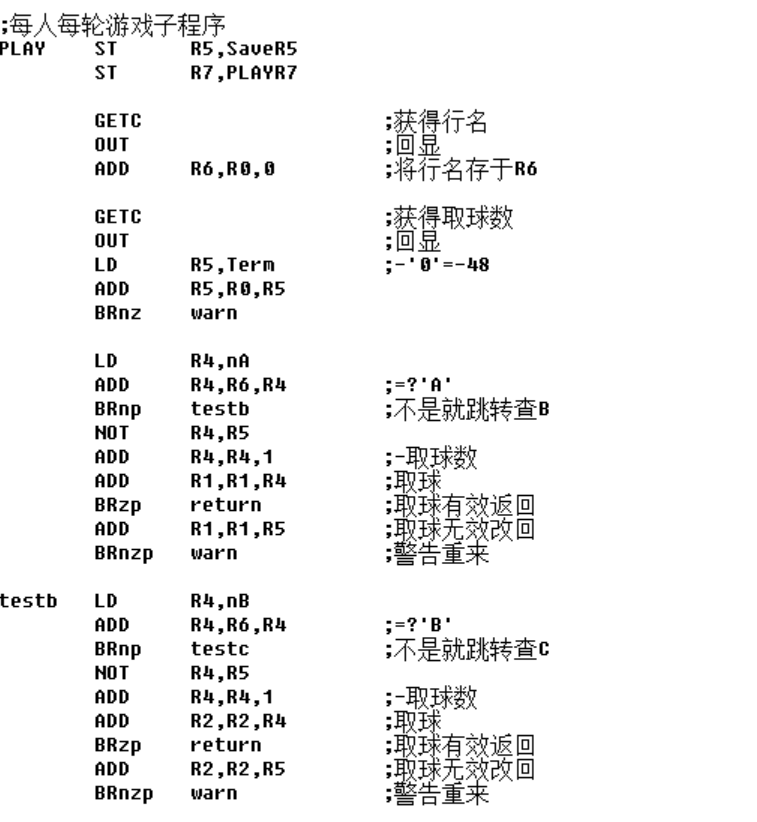
2.界面展示子程序DISPLAY：输出游戏盘面的行提示（ROW A/B/C）及对应行的石头数，通过调用PUTBALL子程序实现石头数的可视化

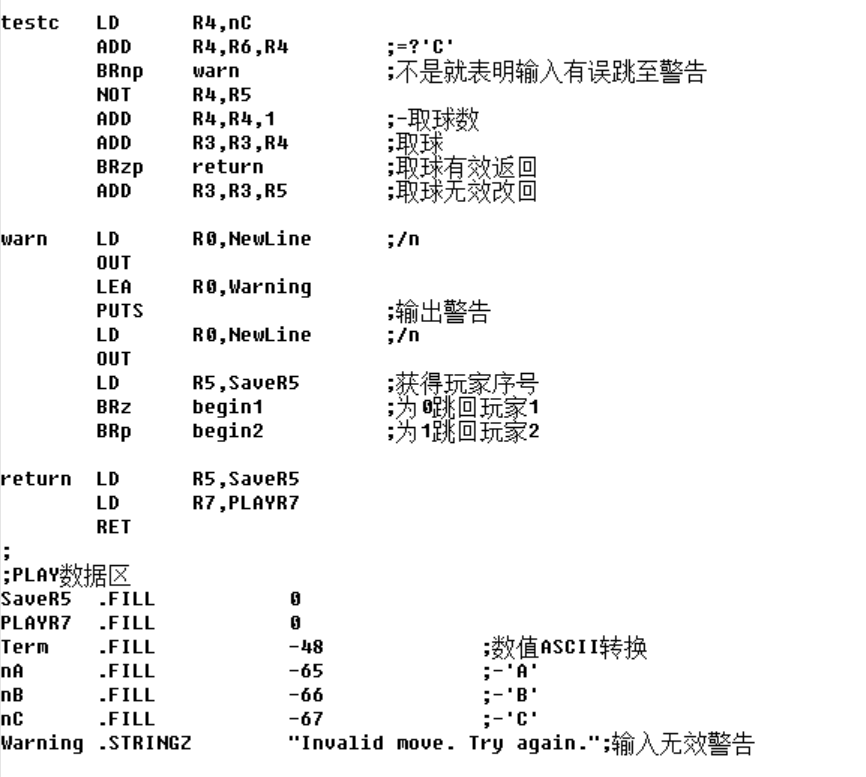


3.球数展示子程序PUTBALL：循环输出球，直至球数减为0。

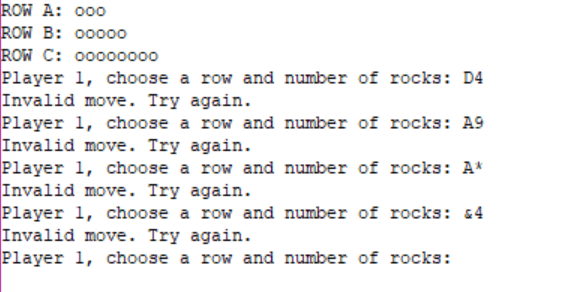


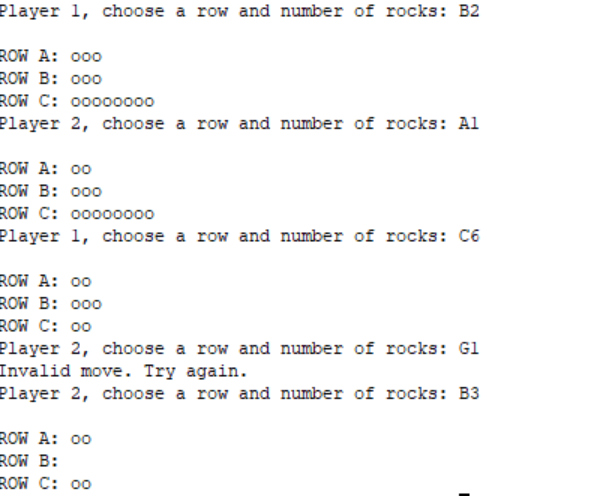
4.单人每轮子程序PLAY：根据输入的行名和球数判断是否有效，无效就警告并重新开始。处理玩家单轮操作，验证行号（A/B/C）和取球数（≤行球数且 > 0），更新盘面，处理错误输入。

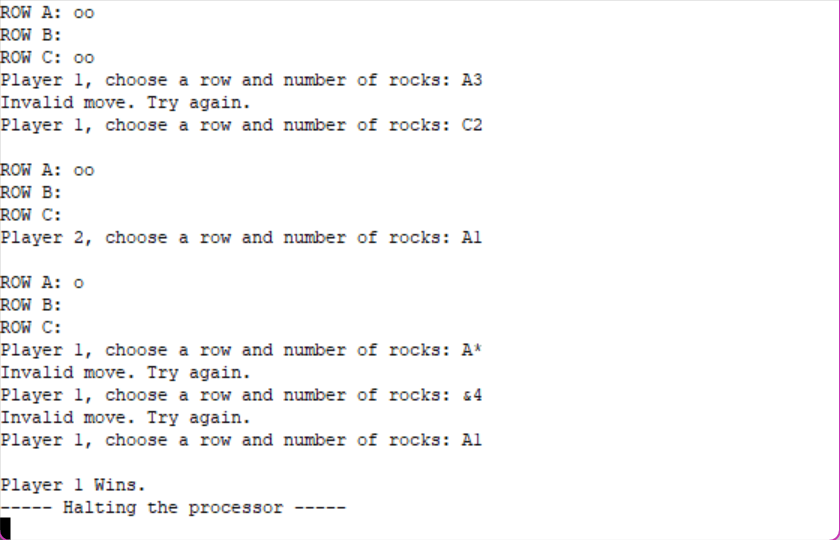




二）样例测试







# 四、实验结论或体会

（撰写实验收获及思考）

运行结果与样例一致，程序设计正确。

通过本次实验尝试运用LC3实现Nim游戏，首次尝试运用子程序结构进行代码编写，同时完成了界面实现。在该过程中我意识到对返回地址的保护（调用者保护）的重要性，尤其在嵌套调用子程序时。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。