**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机系统1**  **项目名称 Nim游戏**  **学 院 计算机与软件学院**  **专 业 计算机技术与科学**  **指导教师 陈飞**  **报 告 人 邓瑞霖 学号 2024150040**  **实验时间 2025年 5 月 29 日**  **提交时间 2024年 6 月 13 日** |

**教务处制**

# 实验目的与要求

使用LC-3仿真器写出Nim游戏程序

包含每轮显示各自的剩余石头数量：

*ROW A: ooo*

*ROW B: ooooo*

*ROW C: oooooooo*

提示用户输入

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

输入不可用时输出错误信息

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

*Invalid move. Try again.*

如果玩家2移除了最后的石头，程序应该输出一下内容：

*Player 2 Wins.*

# 二、实验内容与方法

在本实验中，将使用LC-3的jsr指令，编写子函数，以达到简化编写的目的。

# 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）

1. **概述**

将本实验分割成4个函数，分别负责输出abc球的数量信息SHOW，Player1的逻辑判断，Player2的逻辑判断，输入判断。

1. **初始化数组**

AND R1,R1,#0;

ADD R1,R1,#3;R1初始值为3，存放ROWA棋子的数量

AND R2,R2,#0;

ADD R2,R2,#5;R2初始值为5，存放ROWB棋子的数量

AND R3,R3,#0;

ADD R3,R3,#8;R3初始值为8，存放ROWC棋子的数量

1. **编写SHOW 子程序，输出当前游戏状态**

;子程序SHOW代码：子程序SHOW的作用是输出游戏状态

SHOW ST R1,STORER1;把R1的值先存进内存

ST R2,STORER2;

ST R3,STORER3;

ST R7,STORER7;

;先输出ROWA

SHOWA LEA R0,ROWA

TRAP x22

JUDGER1 ADD R1,R1,#0

BRz SHOWB

LEA R0,SHOWO

TRAP X22

ADD R1,R1,#-1

BRnzp JUDGER1

;输出ROWB

SHOWB LD R0,NEXTLINE

TRAP x21

LEA R0,ROWB

TRAP x22

JUDGER2 ADD R2,R2,#0

BRz SHOWC

LEA R0,SHOWO

TRAP x22

ADD R2,R2,#-1

BRnzp JUDGER2

;输出ROWC

SHOWC LD R0,NEXTLINE

TRAP x21

LEA R0,ROWC

TRAP x22

JUDGER3 ADD R3,R3,#0

BRz SHOWOUT

LEA R0,SHOWO

TRAP x22

ADD R3,R3,-1

BRnzp JUDGER3

SHOWOUT LD R0,NEXTLINE

TRAP x21

LD R1,STORER1;还原R1的值

LD R2,STORER2

LD R3,STORER3

LD R7,STORER7

RET

;子程序SHOW数据区

SHOWO .STRINGZ "o"

STORER1 .FILL 0

STORER2 .FILL 0

STORER3 .FILL 0

STORER7 .FILL 0

NEXTLINE .FILL x000A

ROWA .STRINGZ "ROW A: "

ROWB .STRINGZ "ROW B: "

ROWC .STRINGZ "ROW C: "

1. **编写Player1 的程序判断**

;子程序P1代码：子程序P1的作用是轮到选手1输入，判断输入是否正确，直到输入正确为止

P1 ST R7,P1STORER7

AGAINP1 LEA R0,PLAYER1

TRAP x22

TRAP x20

TRAP x21

ADD R4,R0,#0;将输入的第一个字符存入R4，即行号

TRAP x20

TRAP x21

ADD R5,R0,#0;将输入的第二个字符存入R5,即需要减少的棋子数

LD R0,NEXTLINEP1

TRAP x21

JSR P3

ADD R6,R6,#0;如果输出错误则持续输出

BRp OUTP1

BRnzp AGAINP1

OUTP1 LD R0,NEXTLINEP1

TRAP x21

LD R7,P1STORER7

RET

;子程序P1数据区

P1STORER7 .FILL 0

NEXTLINEP1 .FILL x000A

PLAYER1 .STRINGZ "Player 1, choose a row and number of rocks: "

1. **编写Player2 的程序判断**

;子程序P2代码：子程序P2的作用是轮到选手2输入，判断输入是否正确，直到输入正确为止

P2 ST R7,P2STORER7

AGAINP2 LEA R0,PLAYER2

TRAP x22

TRAP x20

TRAP x21

ADD R4,R0,#0;将输入的第一个字符存入R4，即行号

TRAP x20

TRAP x21

ADD R5,R0,#0;将输入的第二个字符存入R5,即需要减少的棋子数

LD R0,NEXTLINEP2

TRAP x21

JSR P3

ADD R6,R6,#0;如果输出错误则持续输出

BRp OUTP2

BRnzp AGAINP2

OUTP2 LD R7,P2STORER7

RET

;子程序P2数据区

P2STORER7 .FILL 0

NEXTLINEP2 .FILL x000A

PLAYER2 .STRINGZ "Player 2, choose a row and number of rocks: "

1. **编写判断输入是否合法**

;子程序P3代码：子程序P3的作用是判断输入的字母和号码是否合法，合法返回0，不合法返回1 (R6为返回值0或1)

P3 ST R7,P3STORER7

LD R6,A

ADD R6,R6,R4

BRz JUDGER5A

LD R6,B

ADD R6,R6,R4

BRz JUDGER5B

LD R6,C

ADD R6,R6,R4

BRz JUDGER5C

BRnzp FLVAL

JUDGER5A LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL;判断R5是否小于等于0

NOT R6,R1

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R1的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R1,R1,R6

AND R6,R6,#0

ADD R6,R6,#1;输入合法R6为1

BRnzp OUTP3

JUDGER5B LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL

NOT R6,R2

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R2的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R2,R2,R6

AND R6,R6,#0

ADD R6,R6,#1

BRnzp OUTP3

JUDGER5C LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL

NOT R6,R3

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R1的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R3,R3,R6

AND R6,R6,#0

ADD R6,R6,#1

BRnzp OUTP3

FLVAL LEA R0,FALSE

TRAP x22

LD R0,NEXTLINE3

TRAP x21

AND R6,R6,#0;输入不合法R6为0

OUTP3 LD R7,P3STORER7

RET

;子程序P3数据区

P3STORER7 .FILL 0

A .FILL #-65;A的ASC码整数值

B .FILL #-66

C .FILL #-67

ZERO .FILL #-48

NEXTLINE3 .FILL x000A

FALSE .STRINGZ "Invalid move. Try again."

1. **编写主程序**

JSR SHOW

;NEXT1 JSR P1;输入正确后判断是否结束游戏

;

ADD R1,R1,#0

BRnp SHOW1

ADD R2,R2,#0

BRnp SHOW1

ADD R3,R3,#0

BRz OVER1;判断R1,R2,R3是否为0，同时为0结束游戏

SHOW1 LD R0,NEXTLINEMAIN

TRAP x21

JSR SHOW

;

NEXT2 JSR P2;输入正确后判断是否结束游戏

ADD R1,R1,#0

BRnp SHOW2

ADD R2,R2,#0

BRnp SHOW2

ADD R3,R3,#0

BRz OVER2;判断R1,R2,R3是否为0，同时为0结束游戏

SHOW2 LD R0,NEXTLINEMAIN

TRAP x21

JSR SHOW

BRnzp NEXT1

;

OVER1 LEA R0,PLAYER2WIN

TRAP x22

BRnzp OVER

OVER2 LEA R0,PLAYER1WIN

TRAP x22

OVER HALT

;主程序数据区

NEXTLINEMAIN .FILL x000A

PLAYER1WIN .STRINGZ "Player 1 Wins."

PLAYER2WIN .STRINGZ "Player 2 Wins."

;子程序P1代码：子程序P1的作用是轮到选手1输入，判断输入是否正确，直到输入正确为止

1. **将以上所有函数整合，得到最终的程序,最终程序如下**

.ORIG x3000

;R1存放ROWA棋子的数量，R2存放ROWB棋子的数量，R3存放ROWC棋子的数量，R4输入行号，R5输入减少的棋子数

AND R1,R1,#0;

ADD R1,R1,#3;R1初始值为3，存放ROWA棋子的数量

AND R2,R2,#0;

ADD R2,R2,#5;R2初始值为5，存放ROWB棋子的数量

AND R3,R3,#0;

ADD R3,R3,#8;R3初始值为8，存放ROWC棋子的数量

;主程序代码

JSR SHOW

;

NEXT1 JSR P1;输入正确后判断是否结束游戏

;

ADD R1,R1,#0

BRnp SHOW1

ADD R2,R2,#0

BRnp SHOW1

ADD R3,R3,#0

BRz OVER1;判断R1,R2,R3是否为0，同时为0结束游戏

SHOW1 LD R0,NEXTLINEMAIN

TRAP x21

JSR SHOW

;

NEXT2 JSR P2;输入正确后判断是否结束游戏

ADD R1,R1,#0

BRnp SHOW2

ADD R2,R2,#0

BRnp SHOW2

ADD R3,R3,#0

BRz OVER2;判断R1,R2,R3是否为0，同时为0结束游戏

SHOW2 LD R0,NEXTLINEMAIN

TRAP x21

JSR SHOW

BRnzp NEXT1

;

OVER1 LEA R0,PLAYER2WIN

TRAP x22

BRnzp OVER

OVER2 LEA R0,PLAYER1WIN

TRAP x22

OVER HALT

;主程序数据区

NEXTLINEMAIN .FILL x000A

PLAYER1WIN .STRINGZ "Player 1 Wins."

PLAYER2WIN .STRINGZ "Player 2 Wins."

;子程序P1代码：子程序P1的作用是轮到选手1输入，判断输入是否正确，直到输入正确为止

P1 ST R7,P1STORER7

AGAINP1 LEA R0,PLAYER1

TRAP x22

TRAP x20

TRAP x21

ADD R4,R0,#0;将输入的第一个字符存入R4，即行号

TRAP x20

TRAP x21

ADD R5,R0,#0;将输入的第二个字符存入R5,即需要减少的棋子数

LD R0,NEXTLINEP1

TRAP x21

JSR P3

ADD R6,R6,#0;如果输出错误则持续输出

BRp OUTP1

BRnzp AGAINP1

OUTP1 LD R0,NEXTLINEP1

TRAP x21

LD R7,P1STORER7

RET

;子程序P1数据区

P1STORER7 .FILL 0

NEXTLINEP1 .FILL x000A

PLAYER1 .STRINGZ "Player 1, choose a row and number of rocks: "

;子程序P2代码：子程序P2的作用是轮到选手2输入，判断输入是否正确，直到输入正确为止

P2 ST R7,P2STORER7

AGAINP2 LEA R0,PLAYER2

TRAP x22

TRAP x20

TRAP x21

ADD R4,R0,#0;将输入的第一个字符存入R4，即行号

TRAP x20

TRAP x21

ADD R5,R0,#0;将输入的第二个字符存入R5,即需要减少的棋子数

LD R0,NEXTLINEP2

TRAP x21

JSR P3

ADD R6,R6,#0;如果输出错误则持续输出

BRp OUTP2

BRnzp AGAINP2

OUTP2 LD R7,P2STORER7

RET

;子程序P2数据区

P2STORER7 .FILL 0

NEXTLINEP2 .FILL x000A

PLAYER2 .STRINGZ "Player 2, choose a row and number of rocks: "

;子程序P3代码：子程序P3的作用是判断输入的字母和号码是否合法，合法返回0，不合法返回1 (R6为返回值0或1)

P3 ST R7,P3STORER7

LD R6,A

ADD R6,R6,R4

BRz JUDGER5A

LD R6,B

ADD R6,R6,R4

BRz JUDGER5B

LD R6,C

ADD R6,R6,R4

BRz JUDGER5C

BRnzp FLVAL

JUDGER5A LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL;判断R5是否小于等于0

NOT R6,R1

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R1的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R1,R1,R6

AND R6,R6,#0

ADD R6,R6,#1;输入合法R6为1

BRnzp OUTP3

JUDGER5B LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL

NOT R6,R2

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R2的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R2,R2,R6

AND R6,R6,#0

ADD R6,R6,#1

BRnzp OUTP3

JUDGER5C LD R6,ZERO

ADD R6,R6,R5

BRnz FLVAL

NOT R6,R3

ADD R6,R6,#1;

LD R7,ZERO

ADD R6,R6,R7;因为输入的值为ASC码值，R6为R1的相反数，目的是判断棋子数目是否在范围内

ADD R6,R6,R5

BRp FLVAL;如果减去的棋子数目大于原本棋子数目，则发生错误

ADD R6,R5,R7;ASC码值转换为整数值

NOT R6,R6

ADD R6,R6,#1;转为负数让R1减去

ADD R3,R3,R6

AND R6,R6,#0

ADD R6,R6,#1

BRnzp OUTP3

FLVAL LEA R0,FALSE

TRAP x22

LD R0,NEXTLINE3

TRAP x21

AND R6,R6,#0;输入不合法R6为0

OUTP3 LD R7,P3STORER7

RET

;子程序P3数据区

P3STORER7 .FILL 0

A .FILL #-65;A的ASC码整数值

B .FILL #-66

C .FILL #-67

ZERO .FILL #-48

NEXTLINE3 .FILL x000A

FALSE .STRINGZ "Invalid move. Try again."

;子程序SHOW代码：子程序SHOW的作用是输出游戏状态

SHOW ST R1,STORER1;把R1的值先存进内存

ST R2,STORER2;

ST R3,STORER3;

ST R7,STORER7;

;先输出ROWA

SHOWA LEA R0,ROWA

TRAP x22

JUDGER1 ADD R1,R1,#0

BRz SHOWB

LEA R0,SHOWO

TRAP X22

ADD R1,R1,#-1

BRnzp JUDGER1

;输出ROWB

SHOWB LD R0,NEXTLINE

TRAP x21

LEA R0,ROWB

TRAP x22

JUDGER2 ADD R2,R2,#0

BRz SHOWC

LEA R0,SHOWO

TRAP x22

ADD R2,R2,#-1

BRnzp JUDGER2

;输出ROWC

SHOWC LD R0,NEXTLINE

TRAP x21

LEA R0,ROWC

TRAP x22

JUDGER3 ADD R3,R3,#0

BRz SHOWOUT

LEA R0,SHOWO

TRAP x22

ADD R3,R3,-1

BRnzp JUDGER3

SHOWOUT LD R0,NEXTLINE

TRAP x21

LD R1,STORER1;还原R1的值

LD R2,STORER2

LD R3,STORER3

LD R7,STORER7

RET

;子程序SHOW数据区

SHOWO .STRINGZ "o"

STORER1 .FILL 0

STORER2 .FILL 0

STORER3 .FILL 0

STORER7 .FILL 0

NEXTLINE .FILL x000A

ROWA .STRINGZ "ROW A: "

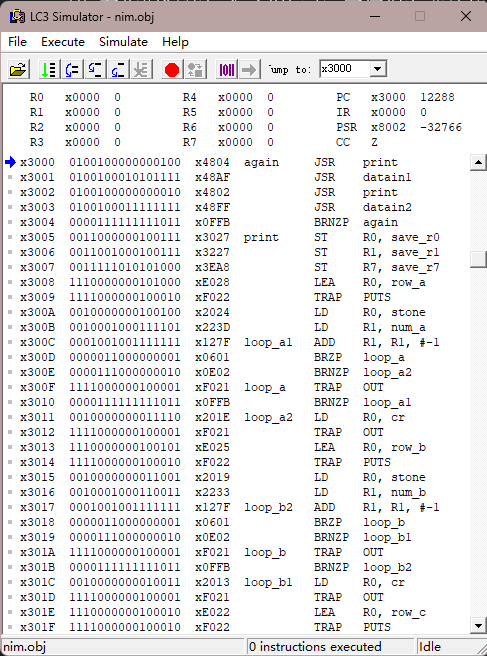
ROWB .STRINGZ "ROW B: "

ROWC .STRINGZ "ROW C: "

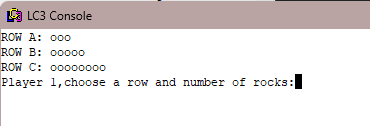
;

.END

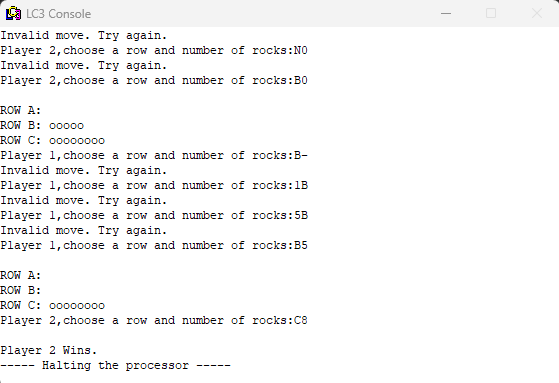
1. **将程序保存命名为 nim.asm,并用simulate打开**



点击运行



正常显示初始石头数量



测试移走石头，报错等功能，均成功。

程序完成

# 四、实验结论或体会

（撰写实验收获及思考）

在本实验中，我编写了一个较为复杂的程序，这不仅进一步锻炼了我的逻辑思维能力，还加深了我对编程技巧的理解。特别地，我在程序中运用了jsr指令来调用子函数，这要求我必须仔细监控r7寄存器的变化。在编写过程中，我发现r7寄存器的内容极易在函数调用过程中被修改，从而影响程序的运行流程，这使我对子函数的调用机制和jsr指令的工作原理有了更深刻的理解。

此外，我在实验中大量使用了字符串输出，这涉及到了lea指令的频繁使用。通过这种方式，我不仅提高了处理字符串和内存地址的能力，也对lea指令的功能和应用有了更全面的掌握。

在实验的执行过程中，我遭遇了一个难题，即一个子函数在返回主函数时出现了逻辑错误。具体表现为程序在执行puts指令前未能妥善保存r7寄存器的值，导致r7寄存器内容被意外改写，子函数无法正确跳回主函数。经过一番调试，我定位到了问题所在，并通过在puts指令执行前添加保存r7寄存器内容的操作来解决了这一问题。这次错误的排查和解决过程，不仅让我对r7寄存器的重要性有了更深入的认识，也提高了我的调试技巧和问题解决能力。

总体来说，通过这次实验，我不仅加深了对jsr和lea指令的理解，还提高了自巑处理复杂逻辑和调试程序的能力。这些经验将对我未来编写更高效、更稳定的程序提供宝贵的帮助。

|  |
| --- |
| 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。