**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 计算机类**  **指导教师 陈飞**  **报 告 人 邓瑞霖 学号 2024150040**  **实验时间 2025年5月29日星期四**  **提交时间 2025年5月30日星期五** |
|  |

**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.*

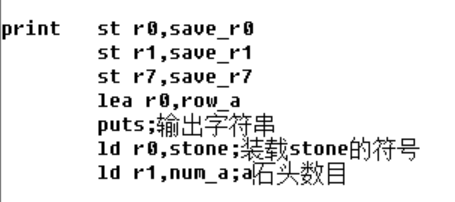
# 二、实验内容与方法

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

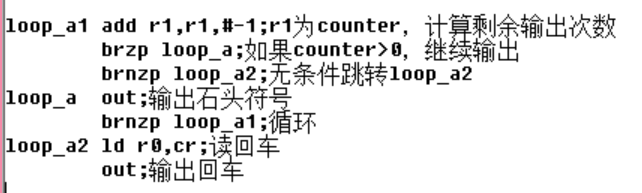
（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

1. 将本实验分割成3个子函数，分别负责输出abc球的数量信息print，player1的数据输入处理datain1，player2的数据输入处理datain2.
2. 编写print函数



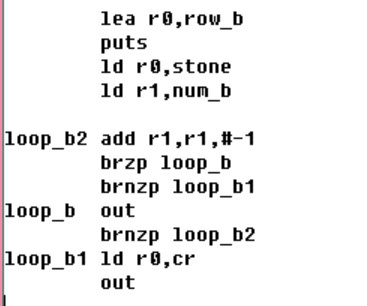
开始先将子函数所需寄存器原数据存起来。

其中，row\_a的数据为row\_a .stringz "ROW A: "。

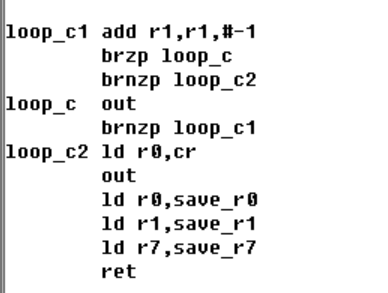


其中，cr的数据cr .fill x000d

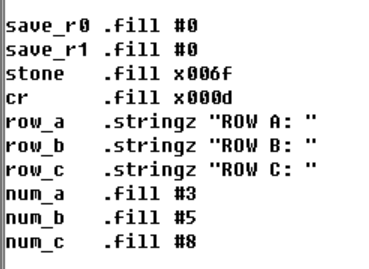
Row a的数据输出完毕，同样方法可以输出Row b和Row c数据



Row b行输出

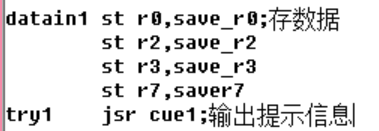


Row c行输出。程序完成，恢复寄存器数据，return回主函数

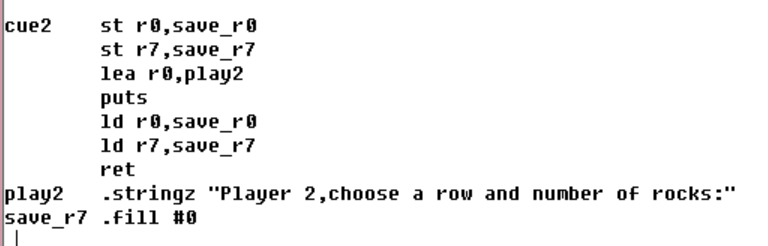


本子函数用到的伪操作，num\_a、num\_b、num\_c存的是初始石头的数量

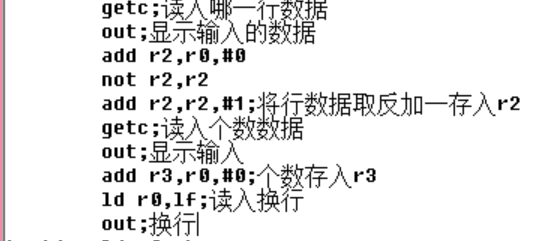
3．编写datain1函数



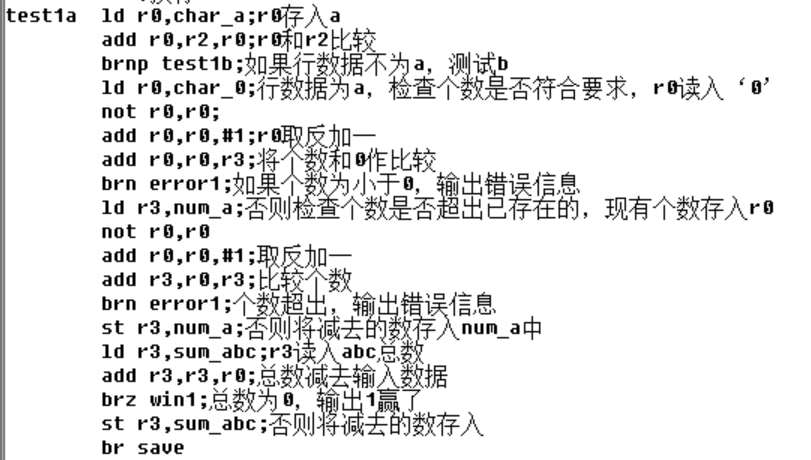
Cue1为输出提示信息用子函数



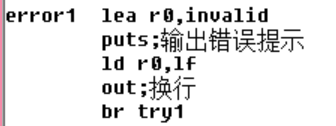
其中r7装载原主函数下一行地址，需储存。输出完成后return回datain1函数。



Lf .fill x000a 完成数据输入后开始处理数据

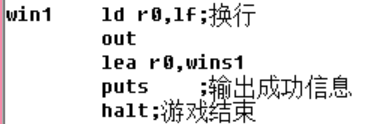


编写error1



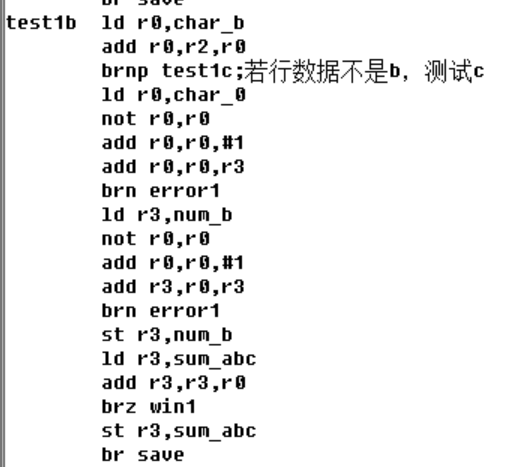
invalid .stringz "Invalid move. Try again."

编写win1

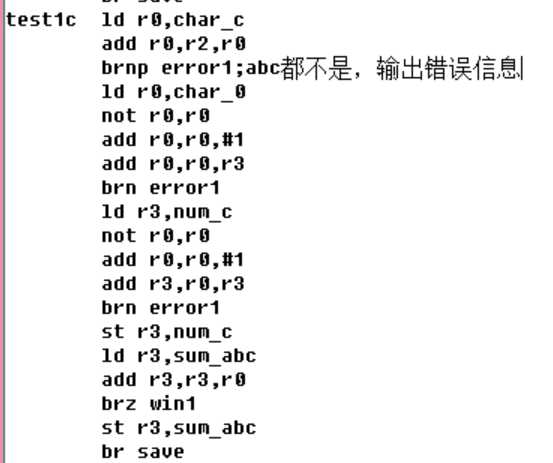


wins1 .stringz "Player 1 Wins."

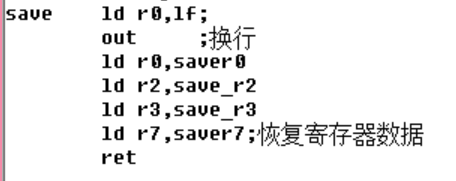
Test1b和test1a相同



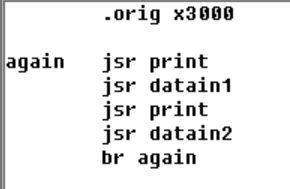
行数据既不是a，也不是b，测试c



Player1在此轮没有赢，回到主函数，进入player2 round。



1. 编写主函数



由以上两个子函数可以写出主函数形式。以上为主函数形式，player1没赢，进入player2 轮，先print。

1. 编写datain2

Datain2形式和datain1相同。直接复制datain1并修改几处数据。如果player2还没赢，循环到开始位置。

Datain2函数如下

cue2 st r0,save\_r0

st r7,save\_r7

lea r0,play2

puts

ld r0,save\_r0

ld r7,save\_r7

ret

play2 .stringz "Player 2,choose a row and number of rocks:"

save\_r7 .fill #0

datain2 st r0,save\_r0

st r2,save\_r2

st r3,save\_r3

st r7,saver7

try2 jsr cue2

getc

out

add r2,r0,#0

not r2,r2

add r2,r2,#1

getc

out

add r3,r0,#0

ld r0,lf

out

test2a ld r0,char\_a

add r0,r2,r0

brnp test2b

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_a

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_a

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

test2b ld r0,char\_b

add r0,r2,r0

brnp test2c

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_b

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_b

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

test2c ld r0,char\_c

add r0,r2,r0

brnp error2

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_c

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_c

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

win2 ld r0,lf

out

lea r0,wins2

puts

halt

error2 lea r0,invalid

puts

ld r0,lf

out

br try2

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

again jsr print

jsr datain1

jsr print

jsr datain2

br again

print st r0,save\_r0

st r1,save\_r1

st r7,save\_r7

lea r0,row\_a

puts;输出字符串

ld r0,stone;装载stone的符号

ld r1,num\_a;a石头数目

loop\_a1 add r1,r1,#-1;r1为counter，计算剩余输出次数

brzp loop\_a;如果counter>0，继续输出

brnzp loop\_a2;无条件跳转loop\_a2

loop\_a out;输出石头符号

brnzp loop\_a1;循环

loop\_a2 ld r0,cr;读回车

out;输出回车

lea r0,row\_b

puts

ld r0,stone

ld r1,num\_b

loop\_b2 add r1,r1,#-1

brzp loop\_b

brnzp loop\_b1

loop\_b out

brnzp loop\_b2

loop\_b1 ld r0,cr

out

lea r0,row\_c

puts

ld r0,stone

ld r1,num\_c

loop\_c1 add r1,r1,#-1

brzp loop\_c

brnzp loop\_c2

loop\_c out

brnzp loop\_c1

loop\_c2 ld r0,cr

out

ld r0,save\_r0

ld r1,save\_r1

ld r7,save\_r7

ret

save\_r0 .fill #0

save\_r1 .fill #0

stone .fill x006f

cr .fill x000d

row\_a .stringz "ROW A: "

row\_b .stringz "ROW B: "

row\_c .stringz "ROW C: "

num\_a .fill #3

num\_b .fill #5

num\_c .fill #8

cue1 st r0,save\_r0

st r7,save\_r7

lea r0,play1

puts

ld r0,save\_r0

ld r7,save\_r7

ret

play1 .stringz "Player 1,choose a row and number of rocks:"

cue2 st r0,save\_r0

st r7,save\_r7

lea r0,play2

puts

ld r0,save\_r0

ld r7,save\_r7

ret

play2 .stringz "Player 2,choose a row and number of rocks:"

save\_r7 .fill #0

datain1 st r0,save\_r0;存数据

st r2,save\_r2

st r3,save\_r3

st r7,saver7

try1 jsr cue1;输出提示信息

getc;读入哪一行数据

out;显示输入的数据

add r2,r0,#0

not r2,r2

add r2,r2,#1;将行数据取反加一存入r2

getc;读入个数数据

out;显示输入

add r3,r0,#0;个数存入r3

ld r0,lf;读入换行

out;换行

test1a ld r0,char\_a;r0存入a

add r0,r2,r0;r0和r2比较

brnp test1b;如果行数据不为a，测试b

ld r0,char\_0;行数据为a，检查个数是否符合要求，r0读入‘0’

not r0,r0;

add r0,r0,#1;r0取反加一

add r0,r0,r3;将个数和0作比较

brn error1;如果个数为小于0，输出错误信息

ld r3,num\_a;否则检查个数是否超出已存在的，现有个数存入r0

not r0,r0

add r0,r0,#1;取反加一

add r3,r0,r3;比较个数

brn error1;个数超出，输出错误信息

st r3,num\_a;否则将减去的数存入num\_a中

ld r3,sum\_abc;r3读入abc总数

add r3,r3,r0;总数减去输入数据

brz win1;总数为0，输出1赢了

st r3,sum\_abc;否则将减去的数存入

br save

test1b ld r0,char\_b

add r0,r2,r0

brnp test1c;若行数据不是b，测试c

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error1

ld r3,num\_b

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error1

st r3,num\_b

ld r3,sum\_abc

add r3,r3,r0

brz win1

st r3,sum\_abc

br save

test1c ld r0,char\_c

add r0,r2,r0

brnp error1;abc都不是，输出错误信息

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error1

ld r3,num\_c

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error1

st r3,num\_c

ld r3,sum\_abc

add r3,r3,r0

brz win1

st r3,sum\_abc

br save

win1 ld r0,lf;换行

out

lea r0,wins1

puts ;输出成功信息

halt;游戏结束

error1 lea r0,invalid

puts;输出错误提示

ld r0,lf

out;换行

br try1

datain2 st r0,save\_r0

st r2,save\_r2

st r3,save\_r3

st r7,saver7

try2 jsr cue2

getc

out

add r2,r0,#0

not r2,r2

add r2,r2,#1

getc

out

add r3,r0,#0

ld r0,lf

out

test2a ld r0,char\_a

add r0,r2,r0

brnp test2b

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_a

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_a

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

test2b ld r0,char\_b

add r0,r2,r0

brnp test2c

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_b

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_b

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

test2c ld r0,char\_c

add r0,r2,r0

brnp error2

ld r0,char\_0

not r0,r0

add r0,r0,#1

add r0,r0,r3

brn error2

ld r3,num\_c

not r0,r0

add r0,r0,#1

add r3,r0,r3

brn error2

st r3,num\_c

ld r3,sum\_abc

add r3,r3,r0

brz win2

st r3,sum\_abc

br save

win2 ld r0,lf

out

lea r0,wins2

puts

halt

error2 lea r0,invalid

puts

ld r0,lf

out

br try2

save ld r0,lf;

out ;换行

ld r0,saver0

ld r2,save\_r2

ld r3,save\_r3

ld r7,saver7;恢复寄存器数据

ret

lf .fill x000a

char\_a .fill x0041

char\_b .fill x0042

char\_c .fill x0043

char\_0 .fill x0030

wins1 .stringz "Player 1 Wins."

wins2 .stringz "Player 2 Wins."

invalid .stringz "Invalid move. Try again."

sum\_abc .fill #16

saver0 .fill #0

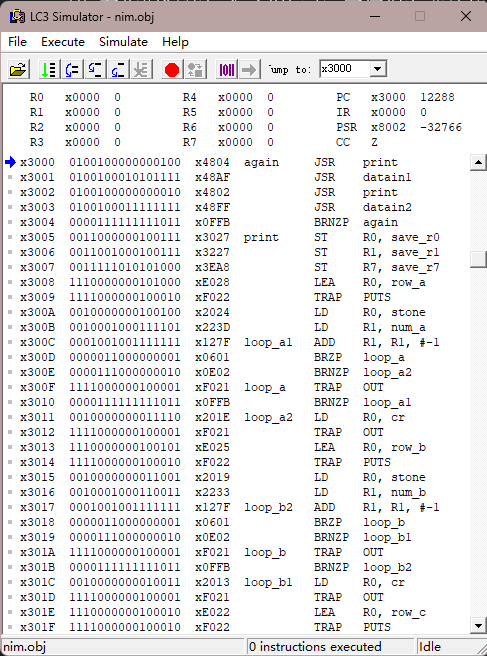
save\_r2 .fill #0

save\_r3 .fill #0

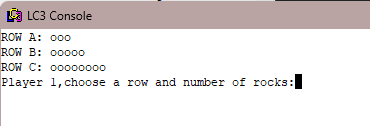
saver7 .fill #0

.end

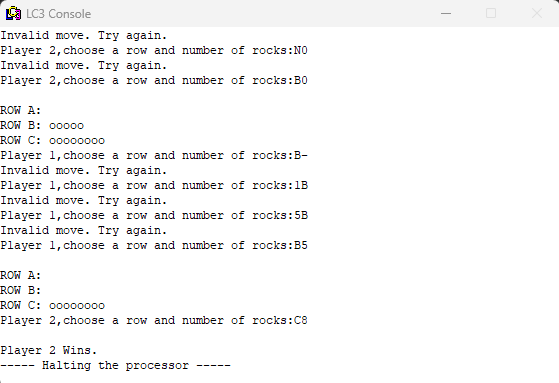
1. 将程序保存命名为 nim.asm,并用simulate打开



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，我写了一个比较复杂的程序，进一步加强了我的逻辑思维能力。我在这个程序使用了jsr指令来写子函数，在实验中须留意被改变的r7寄存器，深化了我对子函数以及jsr指令的原理的认识。同时，本实验中输出提示信息时大量使用字符串输出，深化了我对lea指令的理解。

在本实验中，遇到了一个逻辑错误使得子函数无法跳回主函数。经过调试后发现是有一个子函数在执行puts指令前没有存r7的内容，导致r7被改写无法跳回。经修改后程序顺利运行，这个事件深化了我对保存r7的认识

|  |
| --- |
| 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。