

# NIM游戏

## 1、问题描述

In our variation of Nim, the game board consists of three rows of rocks. Row A contains 3 rocks, Row B contains 5 rocks, and Row C contains 8 rocks. The rules are as follows:

1. Each player takes turns removing one or more rocks from a single row.
2. A player cannot remove rocks from more than one row in a single turn
3. The game ends when a player removes the last rock from the game board. The player who removes the last rock loses.

## 2、实现思路

### 一、初始化需要用到的寄存器，字符

```
AND R0,R0,#0
AND R1,R1,#0;记录A的剩余个数
AND R2,R2,#0;记录B的剩余个数
AND R3,R3,#0;记录C的剩余个数
AND R4,R4,#0;COUNTER
AND R5,R5,#0;记录player的选择行数
AND R6,R6,#0;记录player的选择个数
ADD R1,R1,#3
ADD R2,R2,#5
ADD R3,R3,#8
INFO1 .STRINGZ "ROW A:"
INFO2 .STRINGZ "ROW B:"
INFO3 .STRINGZ "ROW C:"
INFO4 .STRINGZ "o"
INFO5 .STRINGZ "Player 1, choose a row and number of rocks:"
INFO6 .STRINGZ "Player 2, choose a row and number of rocks:"
INFO7 .STRINGZ "Invalid move. Try again."
INFO8 .STRINGZ "Player 1 wins."
INFO9 .STRINGZ "Player 2 wins"
SAVER7 .BLKW #1
A .STRINGZ "A"
B .STRINGZ "B"
C .STRINGZ "C"
ZERO .STRINGZ "0"
SPACE .FILL x000A
NONE .FILL x0000
```

### 二、print部分

PRINT1和PRINT2分别在ROUND1和ROUND2环节被调用，根据R1,R2,R3中储存的信息打印出当前的A,B,C行的数量情况，R4作为counter被调用，用于在打印过程中指示循环的结束。LOOP1表示PRINT1中打印A行里面的数量的循环，LOOP2表示PRINT1中打印B行里面的数量的循环，LOOP3表示PRINT1中打印C行里面的数量的循环，LOOP4表示PRINT2中打印A行里面的数量的循环，LOOP5表示PRINT2中打印B行里面的数量的循环，LOOP6表示PRINT2中打印C行里面的数量的循环。

PRINT1:

```
LEA R0,INFO1
PUTS
AND R4,R4,#0
ADD R4,R4,R1
LOOP1:
ADD R4,R4,#0
BRZ NEXT1
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP1
NEXT1:
LEA R0,SPACE
PUTS
```

```
LEA R0,INFO2
PUTS
AND R4,R4,#0
ADD R4,R4,R2
LOOP2:
ADD R4,R4,#0
BRZ NEXT2
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP2
NEXT2:
LEA R0,SPACE
PUTS
```

```
LEA R0,INFO3
PUTS
AND R4,R4,#0
ADD R4,R4,R3
LOOP3:
ADD R4,R4,#0
BRZ NEXT3
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP3
NEXT3:
LEA R0,SPACE
PUTS
```

BRnzp ROUND1AGAIN

PRINT2:

```
LEA R0,INFO1
PUTS
AND R4,R4,#0
ADD R4,R4,R1
LOOP4:
ADD R4,R4,#0
BRZ NEXT4
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP4
NEXT4:
LEA R0,SPACE
PUTS
```

```
LEA R0,INFO2
PUTS
AND R4,R4,#0
ADD R4,R4,R2
LOOP5:
ADD R4,R4,#0
BRZ NEXT5
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP5
NEXT5:
LEA R0,SPACE
PUTS
```

```
LEA R0,INFO3
PUTS
AND R4,R4,#0
ADD R4,R4,R3
LOOP6:
ADD R4,R4,#0
BRZ NEXT6
LEA R0,INFO4
PUTS
ADD R4,R4,#-1
BRp LOOP6
NEXT6:
LEA R0,SPACE
PUTS
```

BRnzp ROUND2AGAIN

### 三、ROUND1&&ROUND2

ROUND1表示player1走的轮次，ROUND2表示player2走的轮次。在ROUND开始处先调用PRINT函数打印出相应信息，再执行GETC命令读取输入，用R5储存用户输入的行数信息，用R6储存用户输入的个数信息。*这里需要注意的是用户输入的个数在读取的时候是ASCII的模式，需要减去0'的ASCII码，才能得到数字信息。*

```
ROUND1:
BRnzp PRINT1
ROUND1AGAIN:
LEA R0,INFO5
PUTS
GETC
OUT
AND R5,R5,#0
ADD R5,R0,#0
GETC
OUT
AND R6,R6,#0
ADD R6,R0,#0
LEA R0,SPACE
PUTS
AND R4,R4,#0
LD R4,ZERO
NOT R4,R4
ADD R4,R4,#1
ADD R6,R6,R4
```

```
ROUND2:
BRnzp PRINT2
ROUND2AGAIN:
LEA R0,INFO6
PUTS
GETC
OUT
AND R5,R5,#0
ADD R5,R0,#0
GETC
OUT
AND R6,R6,#0
ADD R6,R0,#0
LEA R0,SPACE
PUTS
AND R4,R4,#0
LD R4,ZERO
NOT R4,R4
ADD R4,R4,#1
ADD R6,R6,R4
```

随后，程序需要判断用户输入是否合法，否则输出报错信息，并且让用户重新输入。具体代码如下：（以ROUND1为例）

```
JUDGE1:
AND R4,R4,#0
```

```

LD R4,A
NOT R4,R4
ADD R4,R4,#1
ADD R4,R4,R5
BRZ ROWA
AND R4,R4,#0
LD R4,B
NOT R4,R4
ADD R4,R4,#1
ADD R4,R4,R5
BRZ ROWB
AND R4,R4,#0
LD R4,C
NOT R4,R4
ADD R4,R4,#1
ADD R4,R4,R5
BRZ ROWC
BRnzp ERROR1

```

如果检测到R5储存的是'A'，则跳转到ROWA进行处理，如果检测到R5储存的是'B'，则跳转到ROWB进行处理，如果检测到R5储存的是'C'，则跳转到ROWC进行处理，如果都不是，转向报错信息。

```

ROWA:
AND R4,R4,#0
NOT R4,R6
ADD R4,R4,#1
ADD R4,R4,R1
BRn ERROR1
ADD R1,R4,#0
LEA R0,SPACE
PUTS
BRnzp JUDGEWIN1
F1:
BRnzp ROUND2

ROWB:
AND R4,R4,#0
NOT R4,R6
ADD R4,R4,#1
ADD R4,R4,R2
BRn ERROR1
ADD R2,R4,#0
LEA R0,SPACE
PUTS
BRnzp JUDGEWIN2
F2:
BRnzp ROUND2

ROWC:
AND R4,R4,#0
NOT R4,R6
ADD R4,R4,#1
ADD R4,R4,R3

```

```
BRn ERROR1
ADD R3,R4,#0
LEA R0,SPACE
PUTS
BRnzp JUDGEWIN3
F3:
BRnzp ROUND2
```

检测移走的石头个数是否少于当前行的石头个数，如果是，则跳转到报错信息模块输出报错信息，否则转入判断是否有某一方获胜的模块。并且之后跳转回ROUND2执行。

#### 四、报错信息模块

```
ERROR1:
LEA R0,INFO7
PUTS
LEA R0,SPACE
PUTS
BRnzp ROUND1AGAIN

ERROR2:
LEA R0,INFO7
PUTS
LEA R0,SPACE
PUTS
BRnzp ROUND2AGAIN
```

如果结果不符合要求，输出错误信息，并且退回到输入前重新执行。

#### 五、判断是否获胜模块

```
JUDGEWIN1:
ADD R1,R1,#0
BRnp F1
ADD R2,R2,#0
BRnp F1
ADD R3,R3,#0
BRnp F1
LEA R0,INFO9
PUTS
HALT
```

判断R1,R2,R3是否均为0，如果是，由于调用JUDGE1的是ROUND1模块，说明最后一个石头是player1拿走的，输出player2获胜的信息。结束。

### 3、结果截图

【test1】

```
ROW A:ooo
ROW B:ooooo
ROW C:oooooooo
Player 1, choose a row and number of rocks:A2
```

```
ROW A:o
ROW B:ooooo
ROW C:oooooooo
Player 2, choose a row and number of rocks:B1
```

```
ROW A:o
ROW B:oooo
ROW C:oooooooo
Player 1, choose a row and number of rocks:C3
```

```
ROW A:o
ROW B:oooo
ROW C:ooooo
Player 2, choose a row and number of rocks:A1
```

```
ROW A:
ROW B:oooo
ROW C:ooooo
Player 1, choose a row and number of rocks:B4
```

```
ROW A:
ROW B:
ROW C:ooooo
Player 2, choose a row and number of rocks:C5
```

Player 1 Wins.

--- Halting the LC-3 ---

【test2】

ROW A:ooo  
ROW B:ooooo  
ROW C:ooooooooo  
Player 1, choose a row and number of rocks:B3

ROW A:ooo  
ROW B:oo  
ROW C:ooooooooo  
Player 2, choose a row and number of rocks:C4

ROW A:ooo  
ROW B:oo  
ROW C:oooo  
Player 1, choose a row and number of rocks:A2

ROW A:o  
ROW B:oo  
ROW C:oooo  
Player 2, choose a row and number of rocks:A1

ROW A:  
ROW B:oo  
ROW C:oooo  
Player 1, choose a row and number of rocks:B1

ROW A:  
ROW B:o  
ROW C:oooo  
Player 2, choose a row and number of rocks:C3



ROW A:

ROW B:o

ROW C:o

Player 1, choose a row and number of rocks:B1

ROW A:

ROW B:

ROW C:o

Player 2, choose a row and number of rocks:C1

Player 1 Wins.

--- Halting the LC-3 ---