LAB 04

梁峻滔 PB19051175

1. 算法思想

程序要求实现几个功能：

a) 打印输出每一行的石头

b）①提示 Player 1/2 输入

②接收输入并回显

③检查输入的有效性并打印提示符

④修改石头数

c) 判断游戏是否结束和游戏结束时的赢家

每个功能都需要较多的指令和相对复杂的循环控制才能实现，因此将上述功能拆分成 3 个子程序来实现，再通过循环调用这 3 个子程序来完成游戏。

其中所有的打印输出都用 TRAP x21 实现，键盘输入都用 TRAP x20 实现，所有要输入/输出的字符都须用 R0 存放后才能使用 TRAP x20/21 指令。以下考虑各个功能如何实现。

(1) PrintState 子程序

该子程序实现打印每一行的石头功能。首先需要打印“ROW A/B/C:”字符串，这三个字符串需要通过.STRINGZ 伪指令设置存放在一片连续的内存空间，打印输出时用一个寄存器作为指针和 R0 读入字符后输出即可。问题是打印“ROW A:”之后随即需要打印石头‘o’，打印完 A 行的石头后再接着打印“ROW B:”，

如何衔接起来？这里采用的方法是“ROW A:”“ROW B:”“ROW C:”存在连续的内存空间，这样，用 R1 作为指针，输出完“ROW A:”后 R1 继续+1，然后跳转至打印‘o’的子程序，打印完这一行的‘o’后再跳转回打印行提示符的子程序，这时 R1 已指向“ROW B:”字符串的首地址，就可以顺序打印出来，按同样的方法处理“ROW C:”，同时用 R4 作为一个计数器来判断是否三行都打印完即可。接下来是如何实现打印一定数量的‘o’。首先每一行现有的石头数需要三个内存位置来存放，这里是存放在 x9001、x9002、x9003(实际上可以直接存放在对应的伪指令位置处，但由于敲代码时没有注意到，后面再修改会很麻烦就不做修改了)这三个地址分别存放在 NumA、NumB、NumC 中(用三条连续的.FILL 伪指令指定)。用 R0 存‘o’的 ASCII 码，用另一个指针 R2 指向 NumA/B/C，用 R3 读取 A/B/C 行现有的石头数后作为输出‘o’的计数器，R4 继续作为行的计数器，就可以控制输出哪一行的以及多少个‘o’。检测到当前行石头数为 0 时，则不输出‘o’，另指针指向下一行。

(2) Play 子程序

该子程序实现 part(b)功能。

①提示 Player1/2 输入。使用.STRINGZ 伪指令存放“Player1，choose a row…”“Player2，choose a row…”字符串，该字符串长度为 42(包括‘\0’)，在调用者中预先设置好 Player 标识 R6(为 0 时表示当前是 Player1 操作，为 1 时表示当前是 Player2 操作)，

仍使用 R1 作为字符串指针，先存“Player1，…”的首地址，根据 R6 来判断是否需要令 R1 偏移(指向“Player2,…”)。用 R0 存 R1 指向的字符，再使用 TRAP x21 指令即可输出提示符。

②接收输入并回显。使用 TRAP x20 指令，随后跟一条 TRAP x21 指令即可。由于要输入“A1”“B2”“C6”这样一些大写字母和数字的组合，TRAP 指令只能使用 R0 来接收一个字符，所以要分两次输入，每次输入后都要先把 R0 接收的内容用其他寄存器存起来。

③检查输入的有效性并打印提示符。接收输入后，用输入的大写字母分别与‘A’‘C’比较，有效字母不能大于‘C’，不能小于‘A’，可以用 ASCII 码和 NOT、ADD 指令直接比较；用输入的数字跟指定行现有的石头数比较，输入的石头数不能小于 1，不能大于该行现有的石头数。上述条件只要有一个不满足，该输入就是无效的。输入无效时跳转输出“Invalid…”字符串后重新跳转回开头，然后提示输入、接收输入。

④修改石头数。预先通过间接寻址读入 A 行存放石头数的位置的地址，通过读入的大写字母的 ASCII 码与‘A’的 ASCII 码相减来得到偏移量，指针偏移后读入该行的石头数，对读入的要取走的石头数进行 ASCII 码到数值的转换，相减，判断有效时直接将相减的结果存回该行存放石头数的位置。

(3) IsOver 子程序

该子程序判断游戏是否结束和游戏结束时的赢家。在调用者

中，每次调用完 Play 和 PrintState 后就调用此程序，R6 继续作为输入，标识是哪一个 Player 执行了 Play 程序。游戏是否结束通过判断是否每一行的石头数都为 0 来判断，当游戏结束时 R5 置 1，然后输出赢家的提示符信息；游戏未结束时 R5 置 0。当 R5 返回值为 0 时，在调用者中交换 Player(修改 R6)，然后继续调用 Play 程序。

如此，三个主要的子程序和主程序都搭好了。

2. 代码实现

```
1             .ORIG x3000
2  ;Initialize the number of rocks
3             AND   R0,R0,#0
4             ADD   R0,R0,#3
5             STI   R0,NumA
6             AND   R0,R0,#0
7             ADD   R0,R0,#5
8             STI   R0,NumB
9             AND   R0,R0,#0
10            ADD   R0,R0,#8
11            STI   R0,NumC
12 ;main
13            JSR   PrintState      ;调用 输出面板的子程序
14            AND   R6,R6,#0        ;R6为0表示Player1，为1表示Player2
15 Loop       JSR   Play            ;调用 取石头的子程序
16            JSR   PrintState
17            JSR   IsOver          ;判断游戏是否结束
18            AND   R5,R5,#1        ;R5为IsOver的返回值，为1时表示游戏结束
19            BRp   Finish
20            NOT   R6,R6           ;Player交替
21            AND   R6,R6,#1
22            BRnzp Loop
23 ;Subroutine PrintState
24 PrintState ST    R1,SaveR1
25            ST    R2,SaveR2
26            ST    R3,SaveR3
27            ST    R4,SaveR4
28            ST    R5,SaveR5
29            LD    R0,Newline
30            TRAP  x21
31 ;以下输出"ROW X:"
32            LEA   R1,RowA         ;R1为row指针
33            AND   R4,R4,#0        ;R4作计数器，同时可作为偏移量
34 PrintRow   LDR   R0,R1,#0        ;R0存输入/输出的字符
35            BRz   LDASCII
```

```
36              TRAP  x21
37              ADD   R1,R1,#1
38              BRnzp PrintRow
39  ;以下输出每一行的'o'
40  LDASCII     LEA   R0,ASCII_o
41              LDR   R0,R0,#0      ;R0存'o'的ASCII码
42              LEA   R2,NumA
43              ADD   R2,R2,R4      ;R2存 存有该行现有石头数的位置
44              LDR   R3,R2,#0
45              LDR   R3,R3,#0      ;R3先存好Row A,B或C的石头数量
46  PrintRocks  BRz   RowNoRocks
47              TRAP  x21
48              ADD   R3,R3,#-1     ;R3作输出石头数的计数器
49              BRp   PrintRocks
50  RowNoRocks  LD    R0,Newline    ;输出换行符
51              TRAP  x21
52              ADD   R4,R4,#1      ;R4作为计数器控制输出行
53              ADD   R0,R4,#-3
54              BRz   Return        ;R4=3时结束
55              ADD   R1,R1,#1      ;R1存"ROW B:"或"ROW C:"字符串的首地址，要求A,B,C字符串存放在连续的空间
56              BRnzp PrintRow
57  Return      LD    R1,SaveR1
58              LD    R2,SaveR2
59              LD    R3,SaveR3
60              LD    R4,SaveR4
61              LD    R5,SaveR5
62              RET
63  RowA        .STRINGZ "ROW A:"
64  RowB        .STRINGZ "ROW B:"
65  RowC        .STRINGZ "ROW C:"
66  ASCII_o     .FILL x006F
67  ;Subroutine IsOver
68  IsOver      ST    R1,SaveR1
69              ST    R2,SaveR2
70              ST    R6,SaveR6
```

```
71              AND   R2,R2,#0      ;R2作访问 存放石头数的位置 时的计数器
72              LEA   R1,NumA
73              LDR   R1,R1,#0      ;R1存 存放行A的石头数的地址
74  LDRocks     LDR   R0,R1,#0      ;R0存石头数
75              BRp   FALSE         ;R0为正，说明游戏未结束
76              ADD   R1,R1,#1
77              ADD   R2,R2,#1
78              ADD   R0,R2,#-3
79              BRz   TRUE          ;3行的石头数都是0，游戏结束
80              BRnzp LDRocks
81  TRUE        AND   R5,R5,#0
82              ADD   R5,R5,#1      ;R5置1
83              LEA   R1,Result1
84              LD    R0,Length2
85              AND   R6,R6,#1
86              BRp   PrintWinner   ;R6为1时表示取石头的是Player2，则Player1胜
87              ADD   R1,R1,R0      ;R6为0时Player2胜，R0为偏移量，R1指向2胜的字符串
88  PrintWinner LDR   R0,R1,#0      ;打印胜者信息
89              BRz   Return2
90              TRAP  x21
91              ADD   R1,R1,#1
92              BRnzp PrintWinner
93  FALSE       AND   R5,R5,#0
94  Return2     LD    R1,SaveR1
95              LD    R2,SaveR2
96              LD    R6,SaveR6
97              RET
98  NumA        .FILL x9001
99  NumB        .FILL x9002
100 NumC        .FILL x9003
101 SaveR1      .FILL x0000
102 SaveR2      .FILL x0000
103 SaveR3      .FILL x0000
104 SaveR4      .FILL x0000
105 SaveR5      .FILL x0000
```

```
106  SaveR6       .FILL x0000
107  Newline      .FILL x000A
108  Result1      .STRINGZ "Player1 wins."    ;长14
109  Result2      .STRINGZ "Player2 wins."
110  Length1      .FILL x002A
111  Length2      .FILL x000E
112  ;Print prompt for players,R1 for which player to tell
113  Play         ST    R1,SaveR1
114               ST    R2,SaveR2
115               ST    R3,SaveR3
116               ST    R4,SaveR4
117               ST    R5,SaveR5
118               ST    R6,SaveR6
119               LEA   R1,Player1    ;R1先载入Player1字符串首地址
120               LD    R0,Length1    ;R0载入偏移量
121               AND   R6,R6,#1
122               ST    R1,InnerSave  ;后面的操作会改变R1，需先另存
123               BRz   PrintPrompt   ;R6=0时是Player1，R1不需要偏移
124               ADD   R1,R1,R0      ;R6=1时是Player2，加偏移量后R1指向Player2
125               ST    R1,InnerSave
126  PrintPrompt  LDR   R0,R1,#0
127               BRz   PlayerIn1
128               TRAP  x21
129               ADD   R1,R1,#1
130               BRnzp PrintPrompt
131  PlayerIn1    TRAP  x20           ;输入A,B或C进R0
132  Echo1        TRAP  x21
133               ADD   R2,R0,#0      ;R2存所输入的row的ASCII码
134               LD    R3,ASCII_A
135               ADD   R3,R3,R2      ;R3暂存寻址用的偏移量
136               LEA   R5,NumA
137               ADD   R5,R5,R3      ;R5存NumA,B或C的地址
138               LDR   R5,R5,#0      ;R5=x9001,x9002或x9003
139               LDR   R4,R5,#0      ;R4存该行现有石头数
140  PlayerIn2    TRAP  x20           ;输入要取走的石头数目的ASCII码
```

```
141  Echo2        TRAP  x21
142               LD    R3,ASCII_N
143               ADD   R3,R0,R3      ;R3存输入的石头数的数值
144               BRnz  Error         ;石头数小于1，无效
145               LD    R0,ASCII_A
146               ADD   R0,R2,R0      ;R2跟A的ASCII码比较
147               BRn   Error         ;输入字母小于A，无效
148               LD    R0,ASCII_C
149               ADD   R0,R2,R0      ;R2跟C的ASCII码比较
150               BRp   Error         ;输入字母大于C，无效
151               NOT   R3,R3
152               ADD   R3,R3,#1
153               ADD   R4,R3,R4      ;现有石头数跟要取走的石头数比较
154               BRn   Error         ;要取走大于现有，无效
155               STR   R4,R5,#0      ;更新石头数目
156               LD    R1,SaveR1
157               LD    R2,SaveR2
158               LD    R3,SaveR3
159               LD    R4,SaveR4
160               LD    R5,SaveR5
161               LD    R6,SaveR6
162               RET
163  Error        LD    R0,Newline
164               TRAP  x21
165               LEA   R1,Invalid
166  Errorprompt  LDR   R0,R1,#0      ;输出无效提示符
167               BRz   RecoverR1
168               TRAP  x21
169               ADD   R1,R1,#1
170               BRnzp Errorprompt
171  RecoverR1    LD    R0,Newline
172               TRAP  x21
173               LD    R1,InnerSave  ;恢复R1，以重新打印Player1/2字符串
174               BRnzp PrintPrompt
175  ;
```

```
176  Finish      HALT
177  InnerSave   .FILL x0000
178  ASCII       .FILL x0030
179  ASCII_N     .FILL xFFD0    ;-x0030
180  ASCII_A     .FILL xFFBF    ;-65,-X0041
181  ASCII_C     .FILL xFFBD    ;-67,-X0043
182  Player1     .STRINGZ "Player1,choose a row and number of rocks:"
183  Player2     .STRINGZ "Player2,choose a row and number of rocks:"    ;长42
184  Invalid     .STRINGZ "Invalid.Try again."
185              .END
```

3. 测试

按照所给 Example 输入，得到的结果如下，与 Example 中的过程和结果都一致。

Console (click to focus)

```
ROW A:ooo
ROW B:ooooo
ROW C:oooooooo
Player1,choose a row and number of rocks:B2
ROW A:ooo
ROW B:ooo
ROW C:oooooooo
Player2,choose a row and number of rocks:A1
ROW A:oo
ROW B:ooo
ROW C:oooooooo
Player1,choose a row and number of rocks:C6
ROW A:oo
ROW B:ooo
ROW C:oo
Player2,choose a row and number of rocks:G1
Invalid.Try again.
```

```
Player2,choose a row and number of rocks:B3
ROW A:oo
ROW B:
ROW C:oo
Player1,choose a row and number of rocks:A3
Invalid.Try again.
Player1,choose a row and number of rocks:C2
ROW A:oo
ROW B:
ROW C:
Player2,choose a row and number of rocks:A1
ROW A:o
ROW B:
ROW C:
Player1,choose a row and number of rocks:A*
Invalid.Try again.
Player1,choose a row and number of rocks:&4
```

```
Invalid.Try again.
Player1,choose a row and number of rocks:A1
ROW A:
ROW B:
ROW C:
Player2 wins.

--- Halting the LC-3 ---
```