**实验4：The Game of Nim**

实验目的：用LC-3实现游戏Nim

一、算法设计

根据游戏规则，程序应当包括对局显示、博弈操作、和结束提示三部分。

①显示部分：

利用字符串指令.STRINGZ构造需要用到的字符串，并存放在指定的内存中

②操作部分：

利用指令TRAP x20（GETC）读取玩家操作以及显示指令TRAP x21将玩家操作显示在显示器上。对于输入的结果进行合法性判断，若非法则提示错误信息并重新执行操作程序；若合法则执行相应的操作（切换玩家并调用显示程序将每次操作结果显示在显示器上）

③结束部分：

每次切换玩家时判断当前局势是否符合结束条件（图中已没有石子），若符合，则提示对应玩家获胜；否则继续执行显示程序和操作程序。

二、程序设计

说明：

R1 R2 R3分别存放ROW A、ROW B、ROW C中的剩余石子数

R4是回合指示器：R4为正代表为player1的回合，为负代表player2的回合。

R5 R6均是临时寄存器，它们存放的数据的意义因子程序而异

初始化：

R1 R2 R3分别存放#3、#5、#8

R4存放#1

显示程序DISPLAY：

1、判断R1R2R3是否全部为0，若是，则跳转到结束程序OVER；否则先打印出ROW X（X代指1、2、3），然后将RX的值传递给R5，R0中存入数据x006F，每次打印出“o”R5就减一直至为0。最后打印换行符。

2、判断R4的正负，若为正，则打印player1的提示语；若为负，则提示player2的提示语

3、执行两次getc程序并将读到的ASCII码分别写入地址label为“ROW”和“NUM”的内存中。然后根据玩家一的回合还是玩家2的回合跳转至操作程序P1或P2

操作程序PX（X=1或2）

1、将地址ROW中的数据读取至R6,然后判断其ASCII码是否为#65、#66或#67，若不是，则跳转错误提示；若是，则根据具体的ASCII码跳转至相应的NUM判断程序。

错误提示：

根据当前玩家回合（利用R4）提示错误信息，并回到显示程序的2

2、NUM判断程序：

将地址NUM中的数据存入R6，将玩家选择的行对应的寄存器的值赋给R5。若R6数值大于R5，则跳转至错误提示程序；否则将R5-R6的值赋值给RX，然后跳转至显示程序DISPLAY处

结束程序OVER：

通过判断R4的值确定当前的回合，此时在谁的回合谁就获胜，提示胜利信息并用HALT退出程序。

三、程序检验：（包含错误提示和胜利提示）

