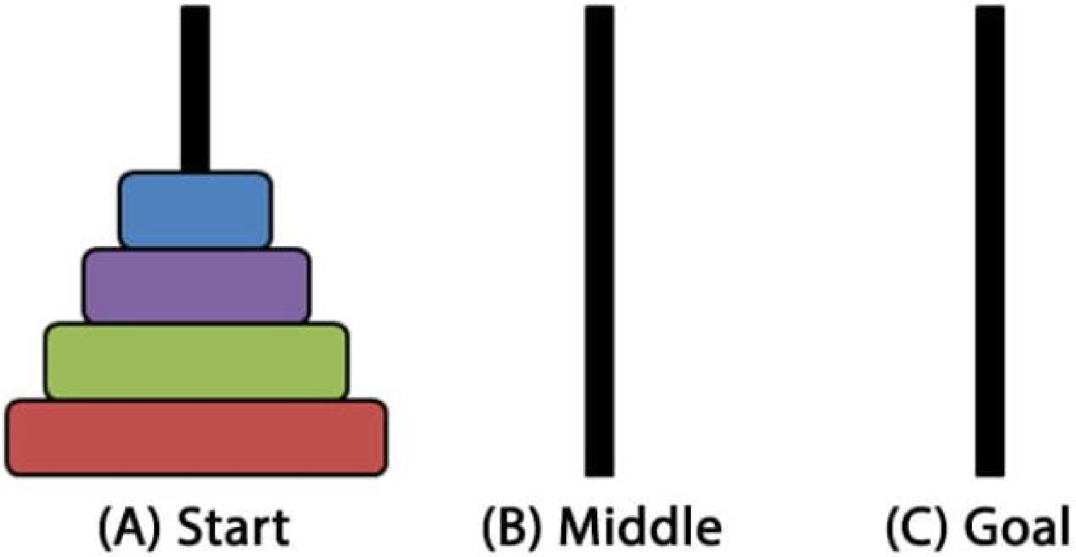
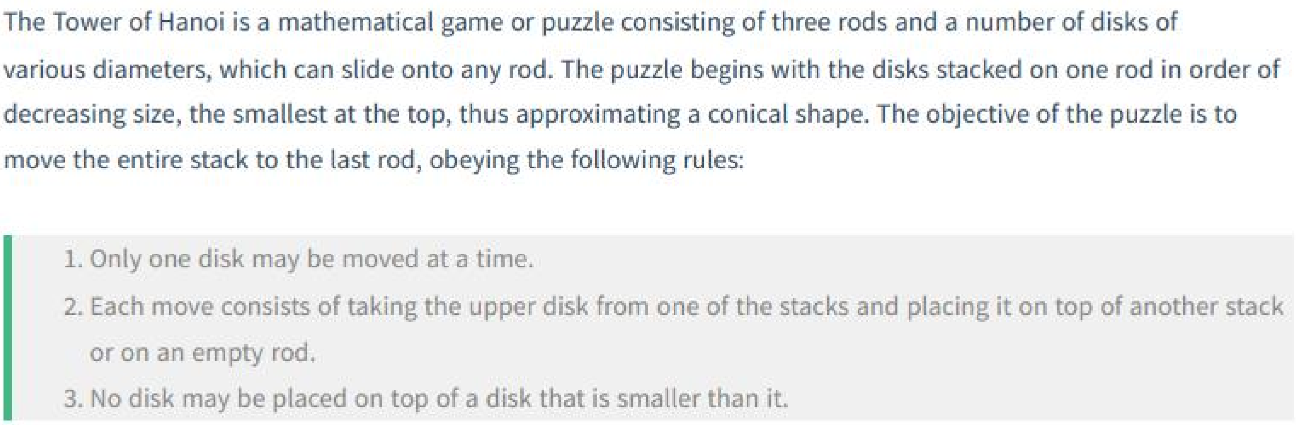
Lab5 实验报告

PB21151807 刘海琳

（一）实验目的

- 本实验将通过使用LC-3 Tools编写汇编码，利用LC-3中的基本指令和寄存器解决以下问题：编写一程序，当键盘没有输入的时候就以合适的速度打印学号。当键盘有输入的时候就判断输入是否是一个十进制数字，并给出相应输出。如果是十进制数字的话就以这个数字为汉诺塔问题的N值，并输出汉诺塔问题的结果。汉诺塔问题描述如下：



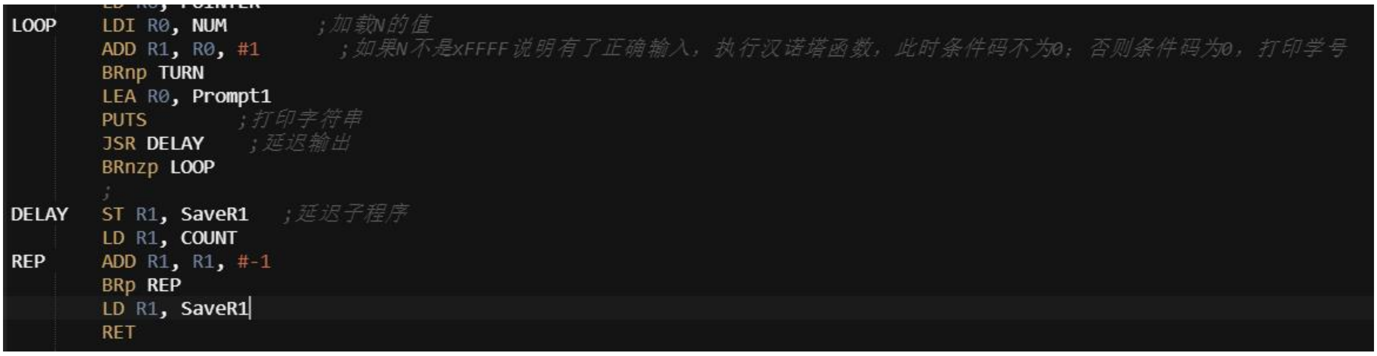


The Tower of Hanoi is a mathematical game or puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a conical shape. The objective of the puzzle is to move the entire stack to the last rod, obeying the following rules:

> 1. Only one disk may be moved at a time.
> 2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
> 3. No disk may be placed on top of a disk that is smaller than it.

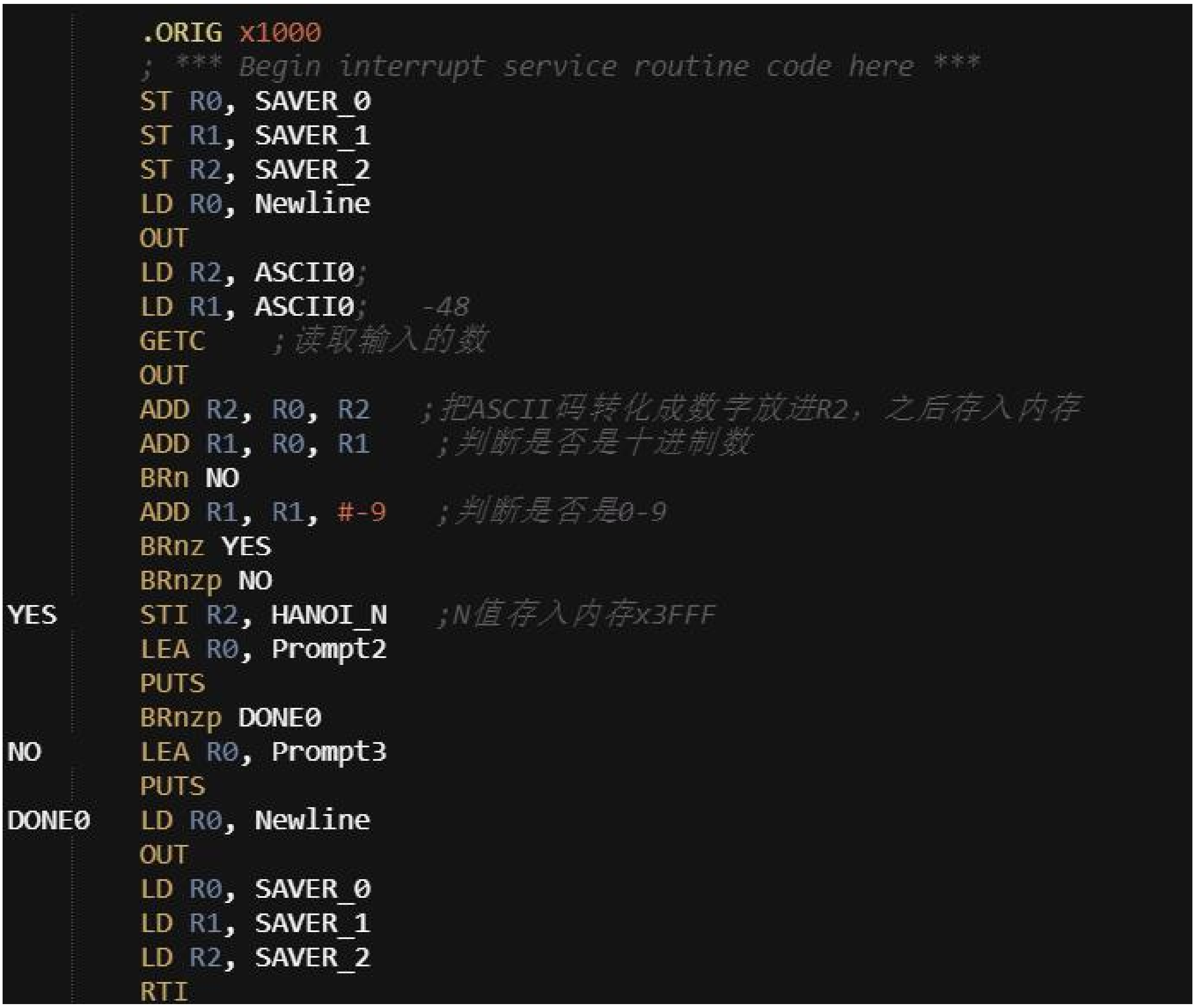
（二）实验原理

- 延迟程序



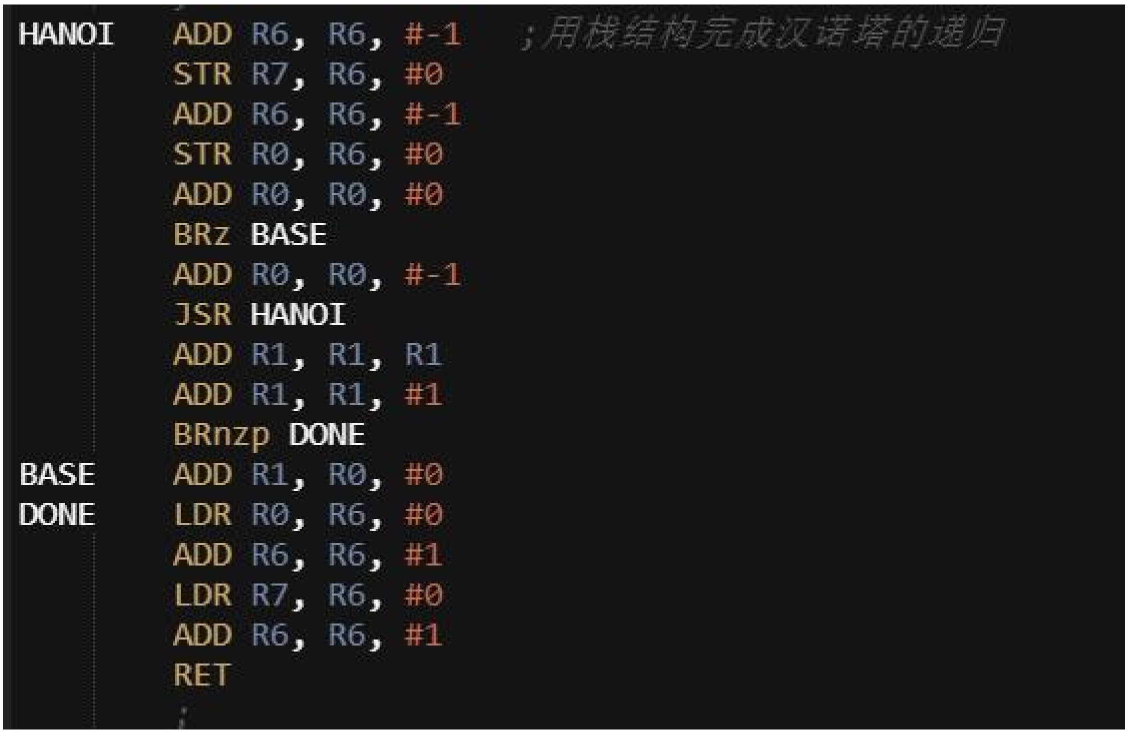
```
LOOP    LDI R0, NUM         ;加载N的值
        ADD R1, R0, #1        ;如果N不是xFFFF说明有了正确输入，执行汉诺塔函数，此时条件码不为0；否则条件码为0，打印学号
        BRnp TURN
        LEA R0, Prompt1
        PUTS          ;打印字符串
        JSR DELAY     ;延迟输出
        BRnzp LOOP
        ;
DELAY   ST R1, SaveR1    ;延迟子程序
        LD R1, COUNT
REP     ADD R1, R1, #-1
        BRp REP
        LD R1, SaveR1
        RET
```

- 判断是否为十进制数字：键盘输入后得到的数其实是ASCII码，减去48后如果是0-9，则说明输入的是十进制数字。直接将减去48后的数字是正数，再减9变成负数，就说明输入的数字是0-9，记得存入x3FFF作为汉诺塔的输入。



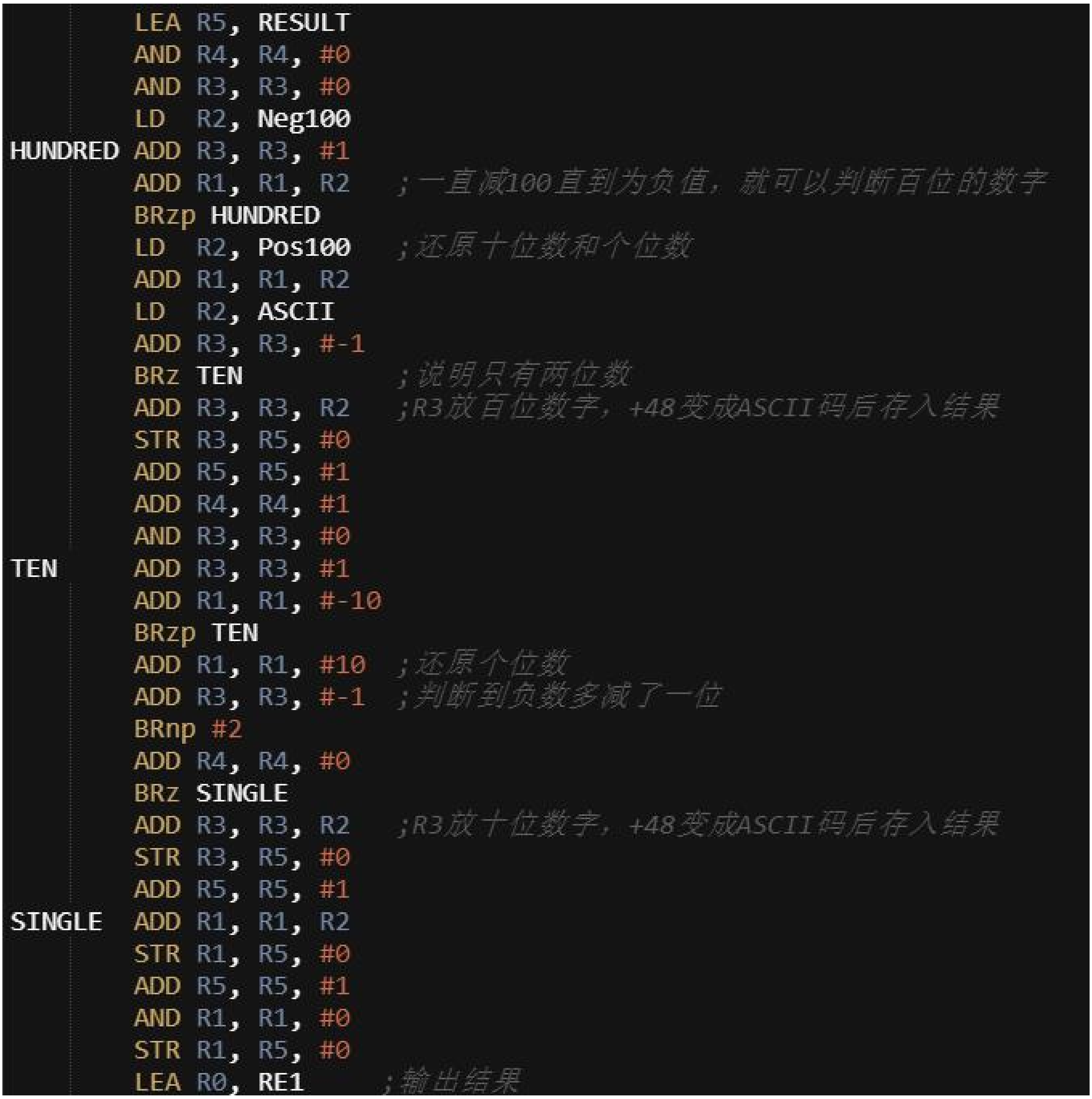
```
        .ORIG x1000
        ; *** Begin interrupt service routine code here ***
        ST R0, SAVER_0
        ST R1, SAVER_1
        ST R2, SAVER_2
        LD R0, Newline
        OUT
        LD R2, ASCII0;
        LD R1, ASCII0;    -48
        GETC    ;读取输入的数
        OUT
        ADD R2, R0, R2   ;把ASCII码转化成数字放进R2，之后存入内存
        ADD R1, R0, R1    ;判断是否是十进制数
        BRn NO
        ADD R1, R1, #-9   ;判断是否是0-9
        BRnz YES
        BRnzp NO
YES     STI R2, HANOI_N   ;N值存入内存x3FFF
        LEA R0, Prompt2
        PUTS
        BRnzp DONE0
NO      LEA R0, Prompt3
        PUTS
DONE0   LD R0, Newline
        OUT
        LD R0, SAVER_0
        LD R1, SAVER_1
        LD R2, SAVER_2
        RTI
```

- 递归的汉诺塔：根据汉诺塔的递归公式H(N)=2H(N-1)+1就可以写出汉诺塔的递归程序。R7是return linkage。



```
HANOI   ADD R6, R6, #-1   ;用栈结构完成汉诺塔的递归
        STR R7, R6, #0
        ADD R6, R6, #-1
        STR R0, R6, #0
        ADD R0, R0, #0
        BRz BASE
        ADD R0, R0, #-1
        JSR HANOI
        ADD R1, R1, R1
        ADD R1, R1, #1
        BRnzp DONE
BASE    ADD R1, R0, #0
DONE    LDR R0, R6, #0
        ADD R6, R6, #1
        LDR R7, R6, #0
        ADD R6, R6, #1
        RET
        ;
```

- 二进制转换成ASCII码：首先先判断百位上的数字，将R1反复减100直至为负值，计数的寄存器再加一，这就是百位的数字。百位数字直接加48转化成数字对应的ASCII码。得到的负值再加回100，就得到了原数字去掉百位后剩下的两位数。再判断十位上的数字，以此类推。



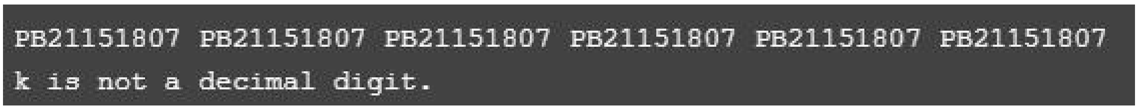
```
        LEA R5, RESULT
        AND R4, R4, #0
        AND R3, R3, #0
        LD  R2, Neg100
HUNDRED ADD R3, R3, #1
        ADD R1, R1, R2    ;一直减100直到为负值，就可以判断百位的数字
        BRzp HUNDRED
        LD  R2, Pos100    ;还原十位数和个位数
        ADD R1, R1, R2
        LD  R2, ASCII
        ADD R3, R3, #-1
        BRz TEN           ;说明只有两位数
        ADD R3, R3, R2    ;R3放百位数字，+48变成ASCII码后存入结果
        STR R3, R5, #0
        ADD R5, R5, #1
        ADD R4, R4, #1
        AND R3, R3, #0
TEN     ADD R3, R3, #1
        ADD R1, R1, #-10
        BRzp TEN
        ADD R1, R1, #10   ;还原个位数
        ADD R3, R3, #-1   ;判断到负数多减了一位
        BRnp #2
        ADD R4, R4, #0
        BRz SINGLE
        ADD R3, R3, R2    ;R3放十位数字，+48变成ASCII码后存入结果
        STR R3, R5, #0
        ADD R5, R5, #1
SINGLE  ADD R1, R1, R2
        STR R1, R5, #0
        ADD R5, R5, #1
        AND R1, R1, #0
        STR R1, R5, #0
        LEA R0, RE1      ;输出结果
```

（三）实验过程

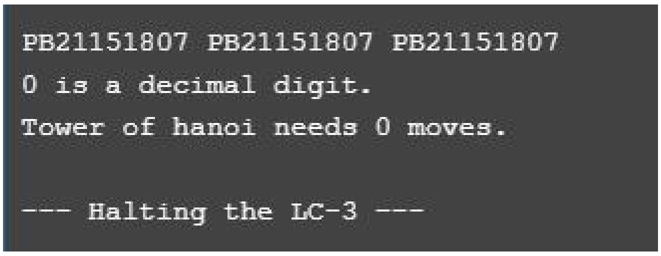
- 汉诺塔部分的程序，在递归时要注意特殊情况输入为0的处理。其次是有时候循环会比递归更简单，这是以后写程序要注意的地方。

（四）测试结果

- 测试结果首先应该涵盖十进制数字0-9以及非十进制数字，其次十进制数字中要包含几个不同的值，比如说N=0的时候汉诺塔输出为0，N=4的时候汉诺塔输出为15，N=7的时候汉诺塔输出为127，这样才能保证二进制转化成ASCII码部分的代码全部正常运行。测试结果如下：

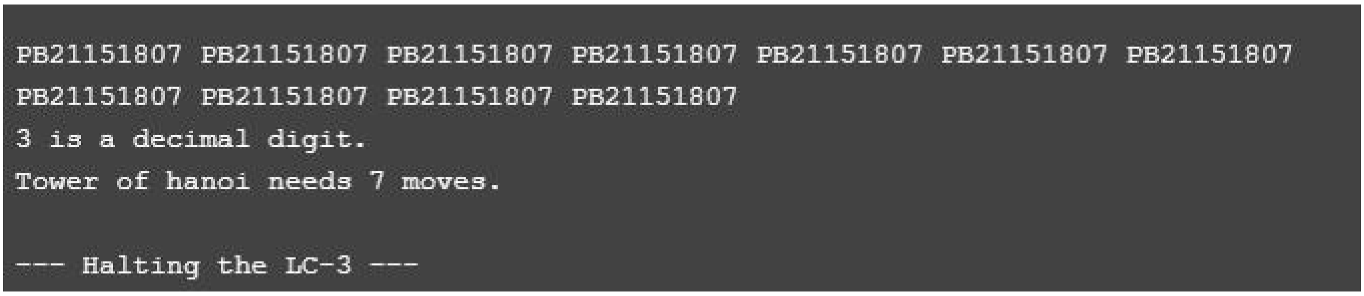


```
PB21151807 PB21151807 PB21151807 PB21151807 PB21151807 PB21151807
k is not a decimal digit.
```



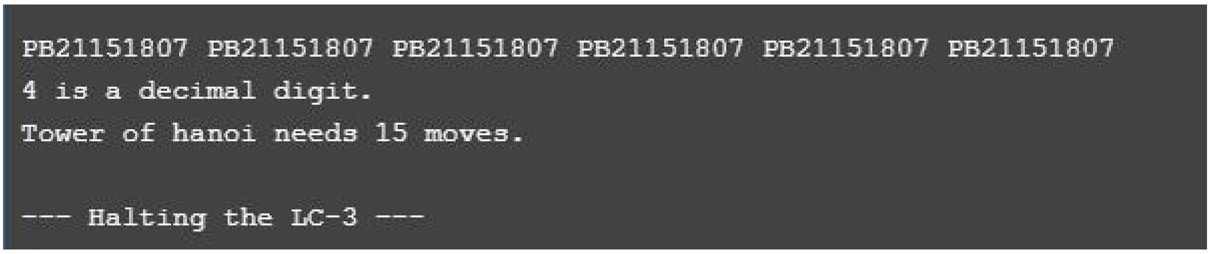
```
PB21151807 PB21151807 PB21151807
0 is a decimal digit.
Tower of hanoi needs 0 moves.

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



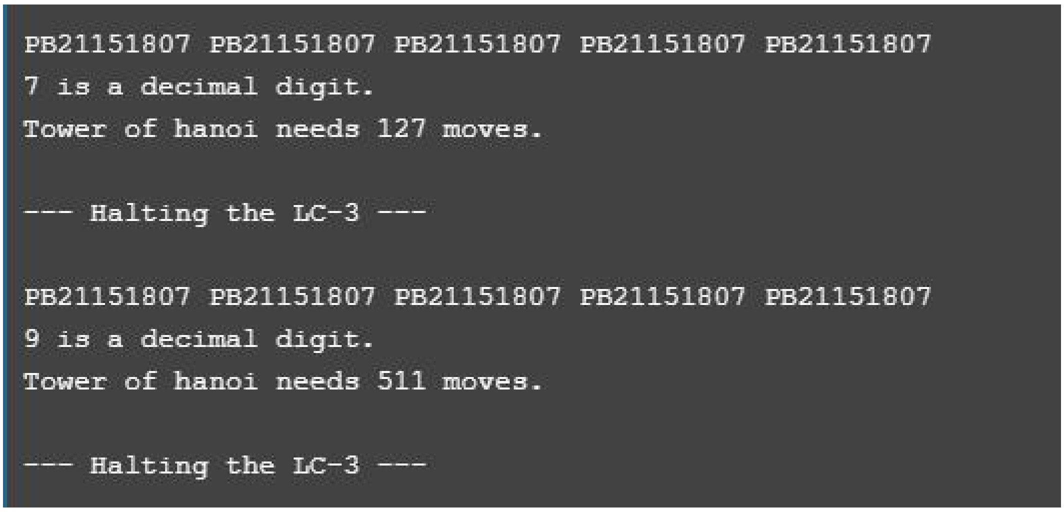
```
PB21151807 PB21151807 PB21151807 PB21151807 PB21151807 PB21151807 PB21151807
PB21151807 PB21151807 PB21151807 PB21151807
3 is a decimal digit.
Tower of hanoi needs 7 moves.

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



```
PB21151807 PB21151807 PB21151807 PB21151807 PB21151807 PB21151807
4 is a decimal digit.
Tower of hanoi needs 15 moves.

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



```
PB21151807 PB21151807 PB21151807 PB21151807 PB21151807
7 is a decimal digit.
Tower of hanoi needs 127 moves.

--- Halting the LC-3 ---

PB21151807 PB21151807 PB21151807 PB21151807 PB21151807
9 is a decimal digit.
Tower of hanoi needs 511 moves.

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

（五）完整代码

```
        .ORIG x800
        ; (1) Initialize interrupt vector table.
        LD R0, VEC
        LD R1, ISR
        STR R1, R0, #0

        ; (2) Set bit 14 of KBSR.
        LDI R0, KBSR
        LD R1, MASK
        NOT R1, R1
        AND R0, R0, R1
        NOT R1, R1
        ADD R0, R0, R1
        STI R0, KBSR

        ; (3) Set up system stack to enter user space.
        LD R0, PSR
        ADD R6, R6, #-1
        STR R0, R6, #0
        LD R0, PC
        ADD R6, R6, #-1
        STR R0, R6, #0
        ; Enter user space.
        RTI

VEC     .FILL x0180
ISR     .FILL x1000
KBSR    .FILL xFE00
MASK    .FILL x4000
PSR     .FILL x8002
PC      .FILL x3000
        .END

        .ORIG x3000
        ; *** Begin user program code here ***
        LD R6, POINTER
LOOP    LDI     R0,     NUM        ;加载N的值
        ADD     R1,     R0, #1       ;如果N不是xFFFF说明有了正确输入，执行汉诺塔函数，此时条件码不为0；否则条件码
        BRnp TURN
        LEA R0, Prompt1
        PUTS        ;打印字符串
        JSR DELAY    ;延迟输出
        BRnzp LOOP
        ;
DELAY   ST R1, SaveR1   ;延迟子程序
        LD R1, COUNT
REP     ADD R1, R1, #-1
        BRp REP
        LD R1, SaveR1
        RET
        ;
HANOI   ADD     R6,     R6, #-1   ;用栈结构完成汉诺塔的递归
        STR     R7,     R6, #0
        ADD     R6,     R6, #-1
        STR     R0,     R6, #0
        ADD     R0,     R0, #0
        BRz BASE
        ADD     R0,     R0, #-1
        JSR HANOI
        ADD     R1,     R1,     R1
        ADD     R1,     R1, #1
        BRnzp DONE
BASE    ADD     R1,     R0, #0
DONE    LDR     R0,     R6, #0
        ADD     R6,     R6, #1
        LDR     R7,     R6, #0
        ADD     R6,     R6, #1
        RET
        ;
TURN    JSR     HANOI   ;之后的部分是把R1中的汉诺塔结果从二进制转换成ASCII码输出到显示器上
        LEA     R5,     RESULT
        AND     R4,     R4, #0
        AND     R3,     R3, #0
        LD      R2,     Neg100
HUNDRED ADD     R3,     R3, #1
        ADD     R1,     R1,     R2   ;一直减100直到为负值，就可以判断百位的数字
        BRzp HUNDRED
        LD      R2,     Pos100   ;还原十位数和个位数
        ADD     R1,     R1,     R2
        LD      R2,     ASCII
        ADD     R3,     R3, #-1
        BRz     TEN          ;说明只有两位数
        ADD     R3,     R3,     R2   ;R3放百位数字，+48变成ASCII码后存入结果
        STR R3, R5, #0
        ADD     R5,     R5, #1
        ADD     R4,     R4, #1
        AND     R3,     R3, #0
TEN     ADD     R3,     R3, #1
        ADD     R1,     R1, #-10
        BRzp TEN
        ADD     R1,     R1, #10  ;还原个位数
        ADD     R3,     R3, #-1  ;判断到负数多减了一位
        BRnp #2
        ADD     R4,     R4, #0
        BRz     SINGLE
        ADD     R3,     R3,     R2   ;R3放十位数字，+48变成ASCII码后存入结果
        STR R3, R5, #0
        ADD     R5,     R5, #1
SINGLE  ADD     R1,     R1,     R2
        STR R1, R5, #0
        ADD     R5,     R5, #1
        AND     R1,     R1, #0
        STR R1, R5, #0
        LEA R0, RE1     ;输出结果
        TRAP x22
        LEA R0, RESULT
        TRAP x22
        LEA R0, RE2
        TRAP x22
        TRAP x25
Prompt1 .STRINGZ "PB21151807 "
COUNT   .FILL x7FFF
SaveR1  .BLKW #1
NUM     .FILL x3FFF
POINTER .FILL xFDFF
Pos100  .FILL x0064
Neg100  .FILL xFF9C
RESULT  .BLKW #4
ASCII   .FILL #48
RE1     .STRINGZ "Tower of hanoi needs "
RE2     .STRINGZ " moves."
        ; *** End user program code here ***
        .END

        .ORIG x3FFF
        ; *** Begin hanoi data here ***
HANOIN .FILL xFFFF
        ; *** End hanoi data here ***
        .END

        .ORIG x1000
        ; *** Begin interrupt service routine code here ***
        ST R0, SAVER_0
        ST R1, SAVER_1
        ST R2, SAVER_2
        LD R0, Newline
        OUT
        LD R2, ASCII0;
        LD R1, ASCII0;   -48
        GETC    ;读取输入的数
        OUT
        ADD R2, R0, R2   ;把ASCII码转化成数字放进R2，之后存入内存
        ADD R1, R0, R1    ;判断是否是十进制数
        BRn NO
        ADD R1, R1, #-9   ;判断是否是0-9
        BRnz YES
        BRnzp NO
YES     STI R2, HANOI_N   ;N值存入内存x3FFF
        LEA R0, Prompt2
        PUTS
        BRnzp DONE0
NO      LEA R0, Prompt3
        PUTS
DONE0   LD R0, Newline
        OUT
        LD R0, SAVER_0
        LD R1, SAVER_1
        LD R2, SAVER_2
        RTI
SAVER_0  .BLKW #1
SAVER_1  .BLKW #1
SAVER_2  .BLKW #1
```

```
HANOI_N .FILL x3FFF
Newline .FILL x000A
ASCII0  .FILL #-48
Prompt2 .STRINGZ " is a decimal digit."
Prompt3 .STRINGZ " is not a decimal digit."
        ; *** End interrupt service routine code here ***
        .END
```