

Assignment Report 2

1. Assignment

Write a program starting at location x3000 in LC-3 assembly language that will:

- 1. Upon starting: Print "Type a name and press Enter:" on the screen and then wait for user to input a string. The input will be ended by pressing <Enter> (ASCII code x000A). Each name is a case-sensitive string of length between 1 to 15 characters.*
- 2. After the user pressed the <Enter> key, your program will search the directory to find the room number based on that name. The address of the first node of the directory has been already stored in location x4000.*
- 3. For each faculty whose first name or last name is the same as input, print his/her full-name and room number in one row. If none exists, print " No Entry" . Each line should be terminated by a carriage return (ASCII x0A).*
- 4. Clean up: Halt the machine.*

2. Solution and Flow Chart

First, I use LEA and TRAP x22 instructions to put out a string of "Type a name and press Enter:" Then I set a loop to read and rewrite chars and save them in the position NAME. Here I used two sub-programs WRITECHAR and READCHAR. Second, after the program detected ENTER (x0A), it goes to the branch SEARCH, which searches the whole linked list to find a match. R6 is used to store whether there is one or more than one matches, so that when there are many matches, the program won't go wrong.

Then, the R0 begins to load its own value (just like a pointer). R2 load the data of names and R1 and R2 begin to compare in a sub-program. If there is a letter that is different then return and R3 is set 0, or R3 is 1. If R3 is 1 then the program goes into OUTPUT and puts out the last name and first name and the room number.

If there is no match and it puts out " No Entry" .

3. Source Code

```
1  .ORIG x3000
2  LEA R0, S1
3  TRAP x22
4  ;
5      LEA R1, NAME
6  INPUT  GETC
7      OUT
8      LD R3, NEWLINE
9      NOT R3, R3
10     ADD R3, R3, #1
11     ADD R3, R3, R0
12     BRz SEARCH
13     STR R0, R1, #0
14     ADD R1, R1, #1
15     BRnzp INPUT
16 ;input the number
17 ;if Enter is detected
18 ;then search the names
19 TRAP x25
20 ;
21 SEARCH
22     ADD R0, R0, xFFF6
23     STR R0, R1, #0
24     LEA R1, NAME
25     LD R0, PTR
26     LDR R0, R0, #0
27     AND R6, R6, #0
28 LOOP
29     LDR R2, R0, #2
30     JSR COMPARE
31     ADD R3, R3, xFFFF
32     BRz OUTPUT
33     LDR R2, R0, #3
34     JSR COMPARE
35     ADD R3, R3, xFFFF
36     BRz OUTPUT
37 HERE LDR R0, R0, #0
38     BRz FAIL
39     BRnzp LOOP
40 ;compare every element
41 ;of the linked list
42 ;if find a match
43 ;then output the names
```

```
44 COMPARE AND R3, R3, #0
45     ADD R3, R3, #1
46     ST R1, SR1
47     ST R2, SR2
48     ST R6, SR6
49 LOOP2 LDR R4, R1, #0
50     LDR R5, R2, #0
51     NOT R4, R4
52     ADD R4, R4, #1
53     ADD R4, R4, R5
54     BRnp NMATCH
55     BRz MATCH
56 NMATCH ADD R3, R3, #-1
57     LD R1, SR1
58     LD R2, SR2
59     LD R6, SR6
60     RET
61 MATCH LDR R6, R1, #0
62     BRz N
63     ADD R1, R1, #1
64     ADD R2, R2, #1
65     BRnzp LOOP2
66 N     LD R1, SR1
67     LD R2, SR2
68     LD R6, SR6
69     RET
70 ;compare each char
71 ;if there is a
72 ;difference then
73 ;return with 0
```

```

74  OUTPUT ADD R6,R6,#1
75      ST R6,SR6
76      AND R6,R6,#0
77      ADD R6,R6,R0
78      LDR R0,R6,#3
79      TRAP x22
80      LEA R0,SPACE
81      TRAP x22
82      LDR R0,R6,#2
83      TRAP x22
84      LEA R0,SPACE
85      TRAP x22
86      LDR R0,R6,#1
87      TRAP x22
88      LD R0,NEWLINE
89      TRAP x21
90      AND R0,R0,#0
91      ADD R0,R0,R6
92      LD R6,SR6
93      BRnzp HERE
94  ;put out the names
95  FAIL
96      ADD R6,R6,#0
97      BRp EXIT
98      LEA R0,S2
99      TRAP x22
100     LD R0,NEWLINE
101     TRAP x21
102     BRnzp EXIT
103 ;if there is no match
104 ;R6 = 0 ,then output
105 ;No Entry

```

```

106 DSR .FILL xFE04
107 DDR .FILL xFE06
108 KBSR .FILL xFE00
109 KBDR .FILL xFE02
110 NEWLINE .FILL x000A
111 PTR .FILL x4000
112 EXIT TRAP x25
113 S1 .STRINGZ "Type a name and press Enter:"
114 S2 .STRINGZ "No Entry"
115 NAME .BLKW 16
116 SR1 .BLKW 1
117 SR2 .BLKW 1
118 SR6 .BLKW 1
119 SPACE .FILL #32
120 ;
121 .END

```

```

.ORIG x3000

LEA R0, S1

TRAP x22

;

    LEA R1,NAME

INPUT    GETC

    OUT

    LD R3,NEWLINE

    NOT R3,R3

    ADD R3,R3,#1

    ADD R3,R3,R0

    BRz SEARCH

    STR R0,R1,#0

    ADD R1,R1,#1

    BRnzp INPUT

;input the number

;if Enter is detected

;then search the names

TRAP x25

;

SEARCH

    ADD R0,R0,xFFFF6

```

STR R0,R1,#0	ADD R3,R3,#1
LEA R1,NAME	ST R1,SR1
LD R0,PTR	ST R2,SR2
LDR R0,R0,#0	ST R6,SR6
AND R6,R6,#0	LOOP2 LDR R4,R1,#0
LOOP	LDR R5,R2,#0
LDR R2,R0,#2	NOT R4,R4
JSR COMPARE	ADD R4,R4,#1
ADD R3,R3,xFFFF	ADD R4,R4,R5
BRz OUTPUT	BRnp NMATCH
LDR R2,R0,#3	BRz MATCH
JSR COMPARE	NMATCH ADD R3,R3,#-1
ADD R3,R3,xFFFF	LD R1,SR1
BRz OUTPUT	LD R2,SR2
HERE LDR R0,R0,#0	LD R6,SR6
BRz FAIL	RET
BRnzp LOOP	MATCH LDR R6,R1,#0
;compare every element	BRz N
;of the linked list	ADD R1,R1,#1
;if find a match	ADD R2,R2,#1
;then output the names	BRnzp LOOP2
COMPARE AND R3,R3,#0	N LD R1,SR1

LD R2,SR2	TRAP x21
LD R6,SR6	AND R0,R0,#0
RET	ADD R0,R0,R6
;compare each char	LD R6,SR6
;if there is a	BRnzp HERE
;difference then	;put out the names
;return with 0	FAIL
OUTPUT ADD R6,R6,#1	ADD R6,R6,#0
ST R6,SR6	BRp EXIT
AND R6,R6,#0	LEA R0,S2
ADD R6,R6,R0	TRAP x22
LDR R0,R6,#3	LD R0,NEWLINE
TRAP x22	TRAP x21
LEA R0,SPACE	BRnzp EXIT
TRAP x22	;if there is no match
LDR R0,R6,#2	;R6 = 0 ,then output
TRAP x22	;No Entry
LEA R0,SPACE	DSR .FILL xFE04
TRAP x22	DDR .FILL xFE06
LDR R0,R6,#1	KBSR .FILL xFE00
TRAP x22	KBDR .FILL xFE02
LD R0,NEWLINE	NEWLINE .FILL x000A

PTR .FILL x4000

EXIT TRAP x25

S1 .STRINGZ "Type a name and press

Enter:"

S2 .STRINGZ "No Entry"

NAME .BLKW 16

SR1 .BLKW 1

SR2 .BLKW 1

SR6 .BLKW 1

SPACE .FILL #32

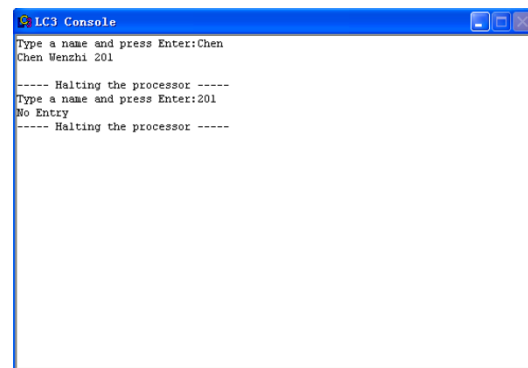
;

.END

本次实验唯一问题就在于直接给了 15 位储存空间，而没有考虑 15 位满的数据后面还要放一个 ‘\0’ 的情况。

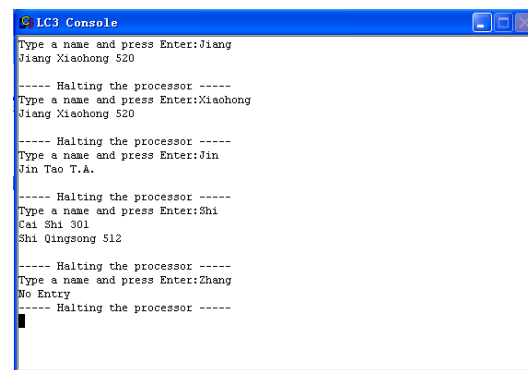
4. Snapshots

Example1 : easy test case :

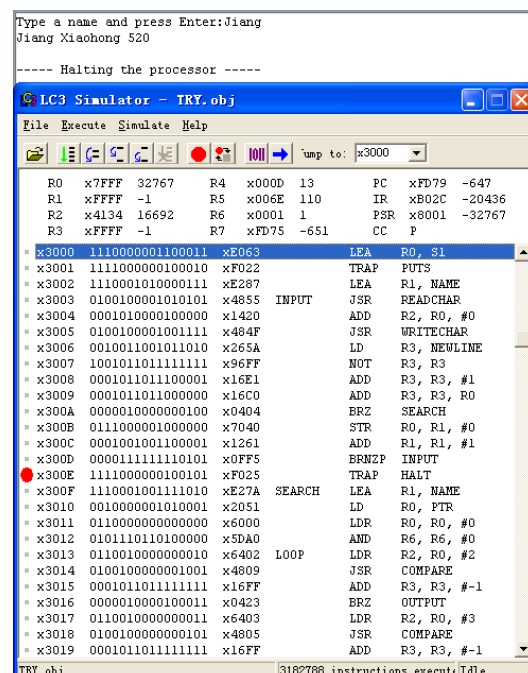


```
LC3 Console
Type a name and press Enter:Chen
Chen Wenzhi 201
----- Halting the processor -----
Type a name and press Enter:201
No Entry
----- Halting the processor -----
```

Example2 : test case from TA :



```
LC3 Console
Type a name and press Enter:Jiang
Jiang Xiaohong 520
----- Halting the processor -----
Type a name and press Enter:Xiaohong
Jiang Xiaohong 520
----- Halting the processor -----
Type a name and press Enter:Jin
Jin Tao T.A.
----- Halting the processor -----
Type a name and press Enter:Shi
Cai Shi 301
Shi Qingsong 512
----- Halting the processor -----
Type a name and press Enter:Zhang
No Entry
----- Halting the processor -----
```



```
LC3 Simulator - TRY.obj
File Execute Simulate Help
Jump to: x3000
R0 x7FFF 32767 R4 x000D 13 PC xFD79 -647
R1 xFFFF -1 R5 x006E 110 IR xB02C -20436
R2 x4134 16692 R6 x0001 1 PSR x8001 -32767
R3 xFFFF -1 R7 xFD75 -651 CC P
x3000 1110000001100011 xE063 LEA R0, S1
x3001 1111000000100010 xF022 TRAP PUTS
x3002 1110001010000111 xE287 LEA R1, NAME
x3003 0100100001010101 x4855 JSR READCHAR INPUT
x3004 0001010000100000 x1420 ADD R2, R0, #0
x3005 0100100001001111 x484F JSR WRITECHAR
x3006 0010011001011010 x265A LD R3, NEWLINE
x3007 1001011011111111 x96FF NOT R3, R3
x3008 0001011011100001 x16E1 ADD R3, R3, #1
x3009 0001011011000000 x16C0 ADD R3, R3, R0
x300A 0000010000000100 x0404 BRZ SEARCH
x300B 0111000001000000 x7040 STR R0, R1, #0
x300C 0001001001100001 x1261 ADD R1, R1, #1
x300D 0000111111110101 x0FF5 BRNZP INPUT
x300E 1111000000100101 xF025 TRAP HALT
x300F 1110001001111010 xE27A LEA R1, NAME SEARCH
x3010 0010000001010001 x2051 LD R0, PTR
x3011 0110000000000000 x6000 LDR R0, R0, #0
x3012 0101110110100000 x5DA0 AND R6, R6, #0
x3013 0110010000000010 x6402 LDR R2, R0, #2 LOOP
x3014 0100100000001001 x4809 JSR COMPARE
x3015 0001011011111111 x16FF ADD R3, R3, #-1
x3016 0000010000100011 x0423 BRZ OUTPUT
x3017 0110010000000011 x6403 LDR R2, R0, #3
x3018 0100100000000101 x4805 JSR COMPARE
x3019 0001011011111111 x16FF ADD R3, R3, #-1
TRY.obj 3182788 instructions executed, Idle
```