

Assignment 5

[Alphabetic.asm](#)

.ORIG x3000

INPUT LDI R5, FFE00

BRzp INPUT

LDI R0, FFE02

REPEAT LDI R5, FFE04

BRzp REPEAT

STI R0, FFE06

JSR ISALPHA

HALT

FFS AND R0, R0, #0

LD R5, MEMR5

LD R4, MEMR4

RET

CHK LD R5, LS

ADD R4, R0, R5

BRn FFS

LD R5, LE

ADD R4, R0, R5

BRp FFS

AND R0, R0, #0

ADD R0, R0, #1

LD R5, MEMR5

LD R4, MEMR4

```
RET
ISALPHA ST R5, MEMR5
    ST R4, MEMR4
    LD R5, US
    ADD R4, R0, R5
    BRn FFS
    LD R5, UE
    ADD R4, R0, R5
    BRp CHK
    AND R0, R0, #0
    ADD R0, R0, #1
    RET
```

```
FFE00 .FILL xFE00
FFE06 .FILL xFE06
FFE04 .FILL xFE04
FFE02 .FILL xFE02
```

```
LS .FILL #-97
US .FILL #-65
UE .FILL #-90
LE .FILL #-122
```

```
MEMR5 .BLKW 1
MEMR4 .BLKW 1
```

.END

Running when input is a letter (Supposed to be 1 at R0)

LC3Tools v2.0.1

Application Edit View

LC3Tools

Registers

R0	x0001	1
R1	xFFA6	65446
R2	xFFE7	65511
R3	x0000	0
R4	xFFE7	65511
R5	x8000	32768
R6	x0000	0
R7	x3007	12295
PSR	x8004	32772 CC: N
PC	x3007	12295
MCR	x0000	0

Console (click to focus)

```
Warning: 103446999: Skipping 'Updating Keyboard' scheduled for 103446990
Warning: 103446999: Skipping 'Updating Display' scheduled for 103446990
Warning: 103446999: Skipping 'No interrupt of higher priority pending' scheduled for 103446991
Warning: 103446999: Skipping 'Triggering pre-instruction callback' scheduled for 103446998
Warning: 105275837: Skipping 'Updating Keyboard' scheduled for 105275830
Warning: 105275837: Skipping 'Updating Display' scheduled for 105275830
Warning: 105275837: Skipping 'No interrupt of higher priority pending' scheduled for 105275831
```

Memory

x3000	xAA21	43553	INPUT LDI R5, FFE00
x3001	x07FE	2046	BRsp INPUT
x3002	xA022	40994	LDI R0, FFE02
x3003	xAA20	43552	REPEAT LDI R5, FFE04
x3004	x07FE	2046	BRsp REPEAT
x3005	xB01D	45085	STI R0, FFE06
x3006	x4810	18448	JSR ISALPHA
x3007	xF025	61477	HALT
x3008	x5020	20512	FFS AND R0, R0, #0
x3009	x2A20	10784	LD R5, MEMR5
x300A	x2820	10272	LD R4, MEMR4
x300B	xC1C0	49600	RET
x300C	x2A19	10777	CHK LD R5, LS
x300D	x1805	6149	ADD R4, R0, R5
x300E	x09F9	2553	BRn FFS
x300F	x2A19	10777	LD R5, LS
x3010	x1805	6149	ADD R4, R0, R5
x3011	x03F6	1014	BRp FFS
x3012	x5020	20512	AND R0, R0, #0
x3013	x1021	4129	ADD R0, R0, #1
x3014	x2A15	10773	LD R5, MEMR5
x3015	x2815	10261	LD R4, MEMR4
x3016	xC1C0	49600	RET

Jump To Location

PC ← ← → →

Running when input is not a letter (Supposed to be 0 at R0)

LC3Tools v2.0.1

Application Edit View

LC3Tools

Registers

R0	x0000	0
R1	xFFA6	65446
R2	xFFE7	65511
R3	x0000	0
R4	xFFE7	65511
R5	x8000	32768
R6	x0000	0
R7	x3007	12295
PSR	x8004	32772 CC: N
PC	x3007	12295
MCR	x0000	0

Console (click to focus)

```
Warning: 103446999: Skipping 'Triggering pre-instruction callback' scheduled for 103446998
Warning: 105275837: Skipping 'Updating Keyboard' scheduled for 105275830
Warning: 105275837: Skipping 'Updating Display' scheduled for 105275830
Warning: 105275837: Skipping 'No interrupt of higher priority pending' scheduled for 105275831
Warning: 108922397: Skipping 'Updating Keyboard' scheduled for 108922390
Warning: 108922397: Skipping 'Updating Display' scheduled for 108922390
Warning: 108922397: Skipping 'No interrupt of higher priority pending' scheduled for 108922391
```

Memory

x3000	xAA21	43553	INPUT LDI R5, FFE00
x3001	x07FE	2046	BRsp INPUT
x3002	xA022	40994	LDI R0, FFE02
x3003	xAA20	43552	REPEAT LDI R5, FFE04
x3004	x07FE	2046	BRsp REPEAT
x3005	xB01D	45085	STI R0, FFE06
x3006	x4810	18448	JSR ISALPHA
x3007	xF025	61477	HALT
x3008	x5020	20512	FFS AND R0, R0, #0
x3009	x2A20	10784	LD R5, MEMR5
x300A	x2820	10272	LD R4, MEMR4
x300B	xC1C0	49600	RET
x300C	x2A19	10777	CHK LD R5, LS
x300D	x1805	6149	ADD R4, R0, R5
x300E	x09F9	2553	BRn FFS
x300F	x2A19	10777	LD R5, LS
x3010	x1805	6149	ADD R4, R0, R5
x3011	x03F6	1014	BRp FFS
x3012	x5020	20512	AND R0, R0, #0
x3013	x1021	4129	ADD R0, R0, #1
x3014	x2A15	10773	LD R5, MEMR5
x3015	x2815	10261	LD R4, MEMR4
x3016	xC1C0	49600	RET

Jump To Location

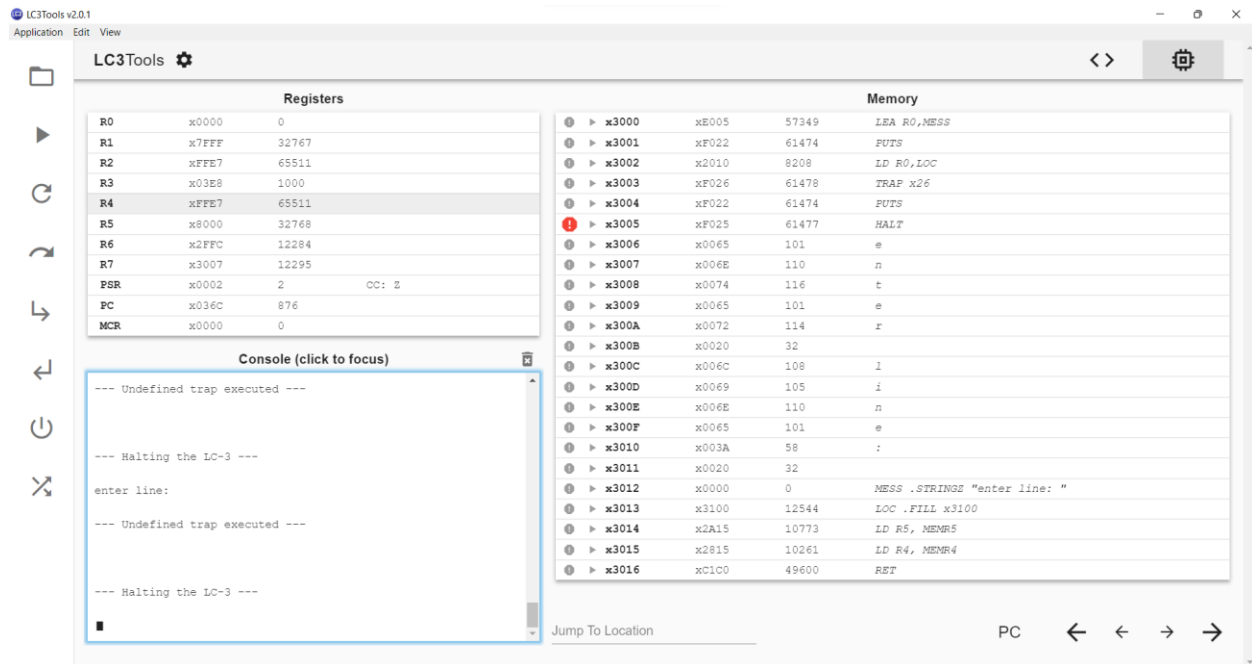
PC ← ← → →

2- .ORIG x3000

```
LEA R0,MESS
PUTS
LD R0,LOC
TRAP x26 ; string STd at x3100
PUTS
HALT
MESS  .STRINGZ "enter line: "
LOC   .FILL x3100
.END
```

```
.ORIG x028A
ADD R4, R0, #0
UI  GETC
OUT
ADD R3, R0, #-10
BRnp KP
RTI
```

```
KP  STR R0, R4, #0
ADD R4, R4, #1
BR UI
.END
```



3-

A) TRAP X72

B) Yes, after the trap instructions, RTI will save the program counter

4-

You can infer from x34 the location (x1000) and from x1000 the address at which the trap instruction is set to start