CS 2110 Timed Lab 3
LC-3 Assembly

Sam Gilson, Cem Gokmen, Henry Harris, Preston Olds

Fall 2018

Contents

1 Before You Begin

Please take the time to read the entire document before starting the assignment. We have made some important updates, and it is your responsibility to follow the instructions and rules.

2 Timed Lab Rules - Please Read

2.1 General Rules

1. You are allowed to submit this timed lab starting at the moment the assignment is released, until you are checked off by your TA as you leave the recitation classroom. Gradescope submissions will remain open until 7:15 pm - but you are not allowed to submit after you leave the recitation classroom under any circumstances. **Submitting or resubmitting the assignment after you leave the classroom is a violation of the honor code - doing so will automatically incur a zero on the assignment and might be referred to the Office of Student Integrity.**
2. Make sure to give your TA your Buzzcard before beginning the Timed Lab, and to pick it up and get checked off before you leave. **Students who leave the recitation classroom without getting checked off or submit after getting checked off will receive a zero.**
3. Although you may ask TAs for clarification, you are ultimately responsible for what you submit. **The information provided in this Timed Lab document takes precedence.** If in doubt, please make sure to indicate any conflicting information to your TAs.
4. Resources you are allowed to use during the timed lab:
   - Assignment files
   - Previous homework and lab submissions
   - Your mind
   - Blank paper for scratch work (please ask for permission from your TAs if you want to take paper from your bag during the Timed Lab)
5. Resources you are **NOT** allowed to use:
   - The Internet (except for submissions)
   - Any resources that are not given in the assignment
   - Textbook or notes on paper or saved on your computer
   - Email/messaging
   - Contact in any form with any other person besides TAs
6. **Before you start, make sure to close every application on your computer.** Banned resources, if found to be open during the Timed Lab period, will be considered a violation of the Timed Lab rules.
7. We reserve the right to monitor the classroom during the Timed Lab period using cameras, packet capture software, and other means.

2.2 Submission Rules

1. Follow the guidelines under the Deliverables section.

2. You are also responsible for ensuring that what you turned in is what you meant to turn in. After submitting you should be sure to download your submission into a brand new folder and test if it works. No excuses if you submit the wrong files, what you turn in is what we grade. In addition, your assignment must be turned in via Gradescope. Under no circumstances whatsoever we will accept any email submission of an assignment. Note: if you were granted an extension you will still turn in the assignment over Gradescope.
3. Do not submit links to files. We will not grade assignments submitted this way as it is easy to change the files after the submission period ends.

2.3 Is collaboration allowed?

Absolutely NOT. No collaboration is allowed for timed labs.

3 Overview

You are given a string of essentially random characters, and your task is to replace all non-numeric characters with space characters.

4 Instructions

The following files have been provided for you:

1. `timedlab3.asm`

You will be editing `timedlab3.asm` to replace instances of non-numeric characters in a string with space characters. The space character (i.e. `' '`) corresponds to decimal value 32 in ASCII. All other ASCII characters in the provided string, outside of the range `'0'...'9'` should be replaced with `' '`.

Remember: Strings are zero-terminated arrays of characters!

An ASCII table has been provided for your reference at the end of this document: The space character is highlighted with red and the numeric digits are highlighted with yellow.

We have defined the following assembly language labels for you:

- `STR_ADDR`: The address of the beginning of the string

Consider the following examples:

Before	After
`"a3hf5k32ss"`	`" 3  5 32  "`
`"A3HF5K32SS"`	`" 3  5 32  "`
`"aBcD3fGhIj"`	`"    3     "`

Note: Replace characters in the existing string. Do not copy it!

Hint: You may create your own labels with `.fill`s!

4.1 Restrictions

You are not allowed to use the `JSR` and `JSRR` instructions for this assignment. In addition, you are not allowed to use Appendix-A or the textbook during this timed lab.

5 Testing Your Work

To test your program, upload `timedlab3.asm` to the Timed Lab 3 assignment on Gradescope. You may resubmit your work as many times as needed, until you sign out and leave the classroom.

6 Common Errors

To trace problems with your code, load the file into `complx`, the LC-3 simulator. To use complx:

1. In the Terminal, type `complx`
2. In the `File` menu, click `Reload`, and open your assembly file (`timedlab3.asm`)
3. Use the `Step` button to run each instruction one step at a time, or use the Run button to execute all of the instructions until `HALT`

7 Rubric

The output of the Gradescope autograder is an approximation of your score on this assignment. The tool is provided so you can evaluate whether your submission fulfills the assignment expectations.

However, we reserve the right to run additional tests, fewer tests, different tests, or potentially change individual tests – your final score will be determined by your instructors, and there is no guarantee your score will correlate with tester output.

8 Deliverables

Please upload the following files to the assignment on Gradescope:

1. `timedlab3.asm`

Do NOT upload an archive, upload the file individually.

Be sure to check your Gradescope test score before you leave the room.

9 LC-3 Assembly Programming Requirements

9.1 Overview

1. Your code must assemble with **NO WARNINGS OR ERRORS**. To assemble your program, open the file with Complx. It will complain if there are any issues. **If the code in this file does not assemble, you WILL get a zero for that file.**
2. **Comment your code!** This is especially important in assembly, because it's much harder to interpret what is happening later, and you'll be glad you left yourself notes on what certain instructions are contributing to the code. Comment things like what registers are being used for and what less intuitive lines of code are actually doing. To comment code in LC-3 assembly just type a semicolon (;), and the rest of that line will be a comment.

3. Avoid stating the obvious in your comments; it doesn't help in understanding what the code is doing. Try to write high-level pseudo-code instead!

   Good Comment

   ```
   ADD R3, R3, -1          ; counter--
   BRp LOOP                ; if counter == 0 don't loop again
   ```

   Bad Comment

   ```
   ADD R3, R3, -1          ; Decrement R3
   BRp LOOP                ; Branch to LOOP if positive
   ```

4. **DO NOT assume that ANYTHING in the LC-3 is already zero.** Treat the machine as if your program was loaded into a machine with random values stored in the memory and register file.

5. Following from 3. You can randomize the memory and load your program by doing File - Randomize and Load.

6. Do not add any comments beginning with @plugin or change any comments of this kind.

7. **Test your assembly.** Don't just assume it works and turn it in.

10 Appendix: ASCII Table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	32	20	SPACE	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	TAB	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL