

Name: **Cameron Peterson-Zopf**

## PROGRAMMING ASSIGNMENT 1

Discussion on a high level with your colleagues is encouraged. Make sure the work submitted is your own. When in doubt, ask a TA or the instructor. If you are not sure what constitutes academic dishonesty, please refer to the AISC web site: <https://aisc.uci.edu/>.

You can fill out your answers below in text, paste screenshots, and/or include images (make sure the image is right side up & legible).

This homework covers:

- LC-3 Machine Code Tutorial

## AISC

Please initial here to indicate you understand UCI's Academic Integrity Policy and confirm that this is your own work you are submitting (this counts for points): **CPZ**

## IDEAL PROGRAM

Here is the code:

```
Lab1.bin
1 ;Our goal is to take the ten numbers which are stored in memory locations x3100 through
2 ;x3109, and add them together, leaving the result in register 1.
3
4 0011 0000 0000 0000 ; start the program at location x3000
5 0101 001 001 1 00000 ; clear R1, to be used for returning the sum
6 0101 100 100 1 00000 ; clear R4, to be used as a counter
7 0001 100 100 1 01010 ; Load R4 with #10, the number of times to add
8 1110 010 011111100 ; Load the starting address of the data
9 0110 011 010 000000 ; Load the next number to be added
10 0001 010 010 1 00001 ; increment the pointer
11 0001 001 001 0 00 011 ; add the next number to the running sum
12 0001 100 100 111111 ; decrement the counter
13 0000 001 11111011 ; do it again if the counter is not yet zero
14 1111 0000 00100101 ; halt
15

attempting to convert C:\Users\camer\OneDrive\UCI Class Topics\Coding\EECS 20\Labs\Lab_1\Lab1.bin into C:\Users\
20\Labs\Lab_1\Lab1.obj
conversion successful
```

Here are the results:

Registers				Memory			
R0	x0000	0		! ▶ x3003	xE4FC	58620	1110010011111100
R1	x0037	55		! ▶ x3004	x6680	26240	0110011010000000
R2	x310A	12554		! ▶ x3005	x14A1	5281	0001010010100001
R3	x000A	10		! ▶ x3006	x1243	4675	0001001001000011
R4	x0000	0		! ▶ x3007	x193F	6463	0001100100111111
R5	x0000	0		! ▶ x3008	x03FB	1019	0000001111111011
R6	x0000	0		! ▶ x3009	xF025	61477	1111000000100101
R7	x0000	0		! ▶ x300A	x0000	0	
PSR	x8002	32770	CC: Z	! ▶ x300B	x0000	0	
PC	x3009	12297		! ▶ x300C	x0000	0	
MCR	x0000	0		! ▶ x300D	x0000	0	

## DEBUG PROGRAM

Here is the initial screen:

```

Lab1P2.bin
1 ;Our goal is to multiply the values in R4 and R5 and store the result in R2.
2
3
4 0011 0010 0000 0000 ; start the program at x3200
5 0101 010 010 1 00000 ; clear R2
6 0001 010 010 0 00 100 ; add R4 to R2, put result in R2
7 0001 101 101 1 1111 ; subtract 1 from R5, put result in R5
8 0000 011 111111101 ; branch to location x3201 if zero or positive
9 1111 0000 00100101 ; halt

attempting to convert C:\Users\camer\OneDrive\UCI Class Topics\Coding\EECS 20\Labs\Lab_1\Lab1P2.bin into C:
20\Labs\Lab_1\Lab1P2.obj
conversion successful

```

Simulator View of Error:

Registers				Memory			
R0	x0000	0		➊ ▶ x3200	x54A0	21664	0101010010100000
R1	x0000	0		➋ ▶ x3201	x1484	5252	0001010010000100
R2	x0014	20		➌ ▶ x3202	x1B7F	7039	0001101101111111
R3	x0000	0		➍ ▶ x3203	x07FD	2045	0000011111111101
R4	x0005	5		❗ ▶ x3204	xF025	61477	1111000000100101
R5	xFFFF	65535		➎ ▶ x3205	x0000	0	
R6	x0000	0		➏ ▶ x3206	x0000	0	
R7	x0000	0		➐ ▶ x3207	x0000	0	
PSR	x8004	32772	CC: N	➑ ▶ x3208	x0000	0	
PC	x3204	12804		➒ ▶ x3209	x0000	0	
MCR	x0000	0		➓ ▶ x320A	x0000	0	

The error is that the branch statement will go back to x3201 if positive and zero. We only want to branch back at positive values. If the counter is zero, we should halt. That is why an extra 5 is being added.

The fixed code is:

```

Lab1P2.bin
1 ;Our goal is to multiply the values in R4 and R5 and store the result in R2.
2
3
4 0011 0010 0000 0000 ; start the program at x3200
5 0101 010 010 1 00000 ; clear R2
6 0001 010 010 0 00 100 ; add R4 to R2, put result in R2
7 0001 101 101 1 11111 ; subtract 1 from R5, put result in R5
8 0000 001 11111101 ; branch to location x3201 result is positive
9 1111 0000 00100101 ; halt

```

The simulator result is:

Registers				Memory			
R0	x0000	0		➤ x3200	x54A0	21664	0101010010100000
R1	x0000	0		➤ x3201	x1484	5252	0001010010000100
R2	x000F	15		➤ x3202	x1B7F	7039	0001101101111111
R3	x0000	0		➤ x3203	x03FD	1021	0000001111111101
R4	x0005	5		➤ x3204	xF025	61477	1111000000100101
R5	x0000	0		➤ x3205	x0000	0	
R6	x0000	0		➤ x3206	x0000	0	
R7	x0000	0		➤ x3207	x0000	0	
PSR	x8002	32770	CC: Z	➤ x3208	x0000	0	
PC	x3204	12804		➤ x3209	x0000	0	
MCR	x0000	0		➤ x320A	x0000	0	