Homework 05

1. An LC-3 assembly language program contains the instruction:
   ASCII LD R1, ASCII
   The label ASCII corresponds to the address x4F08. If this instruction is executed during the running of the program, what will be contained in R1 immediately after the instruction is executed?

   R1 <-- M[ASCII]
   R1 = 0010 001 1 1111 1111
   LD R1, #-1

2. (Adapted from 7.10) The following program fragment has an error in it. Identify the error and explain how to fix it.

	ADD R3, R3, #30	**The immediate value is too large.**
	ST R3, A	
	HALT	
A	.BLKW 1	

   Will this error be detected when this code is assembled or when this code is run on the LC-3?

   The error will be detected by the assembler since it will not be able to form the 16 bits of the instruction which performs the addition. One possible solution is to seperate the addition to two add instruction with immediate of #15.

	ADD R3, R3, #15
	ADD R3, R3, #15
	ST R3, A
	HALT
A	.BLKW 1

3. (Adapted from 7.16) Assume a sequence of nonnegative integers is stored in consecutive memory locations, one integer per memory location, starting at location x4000. Each integer has a value between 0 and 30,000 (decimal). The sequence terminates with the value -1 (i.e., xFFFF).

   a. Create the symbol table entries generated by the assembler when translating the following routine into machine code:

Symbol Table

Label	Memory Address
LOOP	x3003
L1	x300A
NEXT	x300B
DONE	x300D
NUMBERS	x300E

b. What does the above program do?

The instruction AND R2, R1, #1 performs a bit mask (x0001) to decide whether the least significant bit of the value is 0 or 1. The LSB of a number is used to determine whether the integer was even or odd. For example, numbers with a zero LSB are: 0000 (#0), 0010 (#2), 0100 (#4), 0110 (#6), which are all even. Hence, R3 counts the amount of even numbers in the list and R4 counts the amount of odd numbers.

4. (Adapted from 7.18) The following LC-3 program compares two character strings of the same length. The source strings are in the .STRINGZ form. The first string starts at memory location x4000, and the second string starts at memory location x4100. If the strings are the same, the program terminates with the value 1 in R5; otherwise the program terminates with the value 0 in R5. Insert one instruction each at (a), (b), and (c) that will complete the program. Note: The memory location immediately following each string contains x0000.

	.ORIG x3000	
	LD R1, FIRST	
	LD R2, SECOND	
	AND R0, R0, #0	
LOOP	**LDR R3, R1, #0**	(a)
	LDR R4, R2, #0	
	BRz NEXT	
	ADD R1, R1, #1	
	ADD R2, R2, #1	
	NOT R4,R4	(b)
	ADD R4,R4,#1	(c)
	ADD R3, R3, R4	
	BRz LOOP	
	AND R5, R5, #0	
	BRnzp DONE	
NEXT	AND R5, R5, #0	
	ADD R5, R5, #1	
DONE	TRAP x25	
FIRST	.FILL x4000	
SECOND	.FILL x4100	

	.END	

5. The following program does not do anything useful. However, being an electronic idiot, the LC-3 will still execute it.

	.ORIG x3000
	LD R0, Addr1
	LEA R1, Addr1
	LDI R2, Addr1
	LDR R3, R0, #-6
	LDR R4, R1, #0
	ADD R1, R1, #3
	ST R2, #5
	STR R1, R0, #3
	STI R4, Addr4
	HALT
Addr1	.FILL x300B
Addr2	.FILL x000A
Addr3	.BLKW 1
Addr4	.FILL x300D
Addr5	.FILL x300C
	.END

Without using the simulator, answer the following questions:

a. What will the values of registers R0 through R4 be after the LC-3 finishes executing the ADD instruction?

R0 -x300B **R1 -x300D**

R2 -x000A **R3 -x1263**

R4 -x300B

b. What will the values of memory locations Addr1 through Addr5 be after the LC-3 finishes executing the HALT instruction?

Addr1 -x300B **Addr2 -x000A**

Addr3 -x000A **Addr4 -x300B**

Addr5 -x300D

6. The data at memory address x3500 is a bit vector with each bit representing whether a certain power plant in the area is generating electricity (bit = 1) or not (bit = 0). The program counts the number of power plants that generate electricity and stores the result at x3501. However, the program contains a mistake which prevents it from correctly counting the number of electricity generating (operational) power plants. Identify it and explain how to fix it.

	.ORIG x3000	
	AND R0, R0, #0	
	LD R1, NUMBITS	
	LDI R2, VECTOR	
	ADD R3, R0, #1	
CHECK	AND R4, R2, R3	
	BRz NOTOPER	
	ADD R0, R0, #1	
NOTOPER	ADD R3, R3, R3	
	ADD R1, R1, #-1	
	BRp CHECK	
	LD R2, VECTOR	**<-missing instruction**
	STR R0, R2, #1	
	TRAP x25	
NUMBITS	.FILL #16	
VECTOR	.FILL x3500	
	.END	

7. Assemble the folooowing LC-3 assembly language program.

Solution:

The assembled program:

0101 0000 0010 0000 (AND R0, R0, #0)
0001 0100 0010 1010 (ADD R2, R0, #10)
0010 0010 0000 1010 (LD R1, MASK)
0010 0110 0000 1010 (LD R3, PTR1)
0110 1000 1100 0000 (LDR R4, R3, #0)
0101 1001 0000 0001 (AND R4, R4, R1)
0000 0100 0000 0001 (BRz NEXT)
0001 0000 0010 0001 (ADD R0, R0, #1)
0001 0110 1110 0001 (ADD R3, R3, #1)
0001 0100 1011 1111 (ADD R2, R2, #-1)
0000 0011 1111 1001 (BRp LOOP)
1011 0000 0000 0011 (STI R0, PTR2)
1111 0000 0010 0101 (HALT)
1000 0000 0000 0000
0100 0000 0000 0000
0101 0000 0000 0000

This program counts the number of negative values in memory locations 0x4000 - 0x4009 and stores the result in memory location 0x5000.

8. Which is more efficient,interrupt-driven I/O or polling?Explain.

Interrupt-driven I/O is more effcient than polling. Because, in polling, the processor needs to check a specific register (or memory location) regularly to see if anything is being input or output. This consumes unnecessary processing power because the processor checks the register periodically (stopping all other jobs) even when nothing is being input or output. (Most of the time the register will not be inputting or outputting anything unless it is a really I/O-intensive program). However, in interrupt-driven I/O, when something is input or output by a device, the device sends a signal to the processor. Only when the processor receives that signal, it stops all other jobs and does the I/O. Hence, processing power is used for I/O only when it is necessary to do so.

9. (Adapted from 8.15)

   a. What does the following LC-3 program do?

      The keyboard interrupt is enabled, and the digit 2 is repeatedly written to the screen.

   b. If someone strikes a key, the program will be interrupted and the keyboard interrupt service routine will be executed as shown below. What does the keyboard interrupt service routine do?

      The character typed is echoed twice to the screen.

   c. Finally, suppose the program of part (a) started executing, and someone sitting at the keyboard struck a key. What would you see on the screen?

      The digit 2 some number of times, followed by the digit typed twice or three times,followed by the digit 2 continually thereafter.

10. (Adapted from 8.16) What does the following LC-3 program do?

This program outputs ABCDEFGHI.