CIS 350 – INFRASTRUCTURE TECHNOLOGIES
GROUP HOMEWORK #4

Topics: Little Man Computer (LMC) and Assembly Language (Chapter 6)

Worth – 50 points

Write the Group # and Names of the Group Members: Charles Degboe, Anthony Striepe, Daniel Willinger, Karl Dalton

Logistics

1. Get in touch with your group. (See Groups folder on Blackboard.)
2. Discuss and work **all** 3 problems collectively with your group via E-mail, Discussion Forum, Blackboard Collaborate Ultra, and/or MS Teams. (Do not divide the work among the group members. If you collaborate on all problems, you may do better on the tests.)
3. Choose a recorder to prepare the final copy (**one per group**) and submit it via the Blackboard Assignments/Homeworks folder by the due date. You must provide answers on these sheets.
4. Be sure all group members' names are on the final copy. Do **not** add names of your group members who did not participate in the assignment or whose contribution was minimal.

1. Write the LMC program that reads in four numbers (one at a time) and places them in memory locations 22, 23, 24, 25 symbolically denoted by variables *a*, *b*, *c,* and *d*, respectively, in the C# like program segment below. The LMC program should add the second number to the first number, subtract the third number, and add the fourth number as in the formula: (*e:=a+b-c+d*). Next, the program should store the result in memory location 26, symbolically denoted by variable *e*, and print it out. Write the LMC instructions in the mnemonic form. Note that your LMC program will actually implement the following statements written in the C# pseudocode below. The LMC instructions start at address 00.

```
int a, b, c, d, e;
read a, b, c, d;
e:=a+b-c+d;
print e;
```

address	instruction
00	**IN**
01	**STO 22**
02	**IN**
03	**ADD 22**
04	**STO 26**
05	**IN**
06	**SUB 24**
07	**STO 26**
08	**IN**
09	**ADD 26**
10	**STO 26**
11	**LDA 26**
12	**OUT**
13	**HLT**
14	**OUT**
….	
….	
22	**DAT**
23	**DAT**
24	**DAT**
25	**DAT**

26 DAT

What addresses represent the program area and the data area, respectively?

Program area: 00-14 Data area: 22-26

2. Write the LMC program to find the larger of the two numbers stored in memory locations 35 and 36. Write the larger number to memory location 37. To verify if the program runs correctly, you may trace its execution for the following two scenarios:

(a) assume that memory locations 35 and 36 contain 20 and 10, respectively.
(b) assume that memory locations 35 and 36 contain 10 and 20, respectively.

In both scenarios, the program should find the **larger** number, which is 20, and store it in memory location 37. Note that the program starts at address 05.

address	**mnemonic form**
05	**LDA 35**
06	**SUB 36**
07	**BRZ 10**
08	**LDA 35**
09	**STO 37**
10	**LDA 36**
11	**STO 37**
12	**HLT**
13	
14	
15	
16	
17	
….	
…	....
…	....
35	**DAT**
36	**DAT**
37	**DAT**

What addresses represent the program area and the data area, respectively?

Program area: 05-12 Data area: 35-37

3. Below is a Little Man program that solves exercise 6.9, p. 164, from textbook. The program is very similar to the LMC program which you will find in the lecture notes on Chapter 6 posted on Blackboard and discussed in Panopto. The difference is that the program below is somewhat simpler as it uses only 2 branches (BRZ 19 and BR 11), whereas the program in the lecture notes uses 3 branches (BRP 05, BR 10, and BR 01). First, try to understand each instruction thoroughly and then trace the execution of each instruction. Note that the program starts at address 10.

Address	Instruction (Mnemonics)
10	IN
11	STO 25
12	BRZ 19
13	IN
14	ADD 27
15	STO 27
16	LDA 25
17	SUB 26
18	BR 11
19	LDA 27
20	OUT
21	HLT

Address		Contents
25	DAT	? → 2 → 1 → 0 (decremented by 1 each time the loop is executed)
26	DAT	1
27	DAT	0

Assume now that the above program will only read 3 numbers. That is, the following numbers in this order will be placed, one at a time, in the In-basket: 2, 18, and 15, where 2 is the count of numbers that follow, and 18 and 15 are the numbers that are to be added. The first column in the table on page 4 shows the order in which the instructions from the program will be executed. Trace the execution of these instructions and determine the contents of the PC **before** and **after** each instruction is executed. Also, write down in the table the contents of the In-basket; Accumulator; Memory locations 25, 26, and 27; and Out-basket **after** each instruction is executed. Memory location 25 controls the loop. It initially contains an unknown value (?), then 2, next 1, and finally 0. Memory location 26 always contains 1. It is used to decrease the loop count by 1. Memory location 27 is initialized with 0, and finally it stores 33, the sum of 18 and 15. The entry 10 → 11 in the PC column means that the PC is 10 when the instruction IN started and is changed to 11 when the instruction IN is finished.

The sequence in which instructions are executed	PC before → after	In-basket	Accumulator	Memory location 25	Memory location 26	Memory location 27	Out-basket
IN	10 → 11	2	2	?	1	0	?
STO 25	11 -> 12	2	0	2	1	0	?
BRZ 19	12 -> 13	2	0	2	1	0	?
IN	13 -> 14	18	18	2	1	0	?
ADD 27	14 -> 15	18	18	2	1	18	?
STO 27	15 -> 16	18	18	2	1	18	?
LDA 25	16 -> 17	18	2	2	1	18	?
SUB 26	17 -> 18	18	1	2	1	18	?
BR 11	18 -> 11	18	1	2	1	18	?
STO 25	11 -> 12	15	15	2	1	18	?
BRZ 19	12 -> 13	15	33	2	1	18	?
IN	13 -> 14	15	33	2	1	33	?
ADD 27	14 -> 15	15	33	2	1	33	?
STO 27	15 -> 16	15	33	2	1	33	?
LDA 25	16 -> 17	15	2	2	1	33	?
SUB 26	17 -> 18	15	1	2	1	33	?
BR 11	18 -> 19	15	1	2	1	33	3
STO 25	19 -> 20	?	?	2	1	33	3
BRZ 19	20 -> 21	?	?	2	1	33	3
LDA 27							
OUT							
HLT							