CIS 350 – INFRASTRUCTURE TECHNOLOGIES

HOMEWORK # 3

NAME(S): ___Cedric Fenn_____
(You may do this homework in groups of 2 students maximum.)

Topics: Data formats and standards, representing numerical data, and LMC assembly language (Chapters 4-6)

1. Approximately how many pages of pure 16-bit Unicode text can each of the following devices: 8.5 GB DVD-ROM and a 256 GB Solid State Drive (SSD) hold? Assume that a typical page of text holds, say roughly 2500 characters. (You must show your calculations.)

8.5 GB DVD-ROM = **1,825,361.10 pages.**

256 GB Solid State Drive (SSD **= 54,975,581.39 pages.**

2. Use the World Wide Web as a resource to investigate MPEG-2 and MPEG-4 standard. Explain briefly the data compression algorithm used by MPEG-2 and MPEG-4.

MPEG-2 compresses high-definition video data by removing redundant parts from a frame, called lossy compression, to store the frames only once.

MPEG-4 creates high-quality video at a low bit rate by adjusting the bit rate and comparing multiple frames at a time. MPEG-4 can take up less memory while improving audio and video quality.

3. An analog wave representing the sound is sampled with the frequency of 44,100Hz during its conversion to a digital form. Each sample is stored in 2 bytes. How many GB would it take to store 60 minutes of the uncompressed sound? If a compression ratio is 10:1, how many GB would that sound occupy after compression. (You must show your computations.)

0.296 GB
0.0296 GB

4. Find the 16-bit (2-byte) 2's complementary binary representation for the decimal number $(-80)_{10}$. Note that when you convert the 1's complement to 2's complement a carry maybe generated. You must show your computations.

$(1010000)_2$

00000000 01010000

11111111 10101111 + 1

$(11111111\ 10110000)_2$

5. Below is a Little Man program that solves exercise 6.9, p. 192. The program is very similar to the LMC program which you will find in the lecture notes on chapter 6 posted on Blackboard. The difference is that the program below is somewhat simpler as it uses only 2 branches (BRZ 09 and BR 01), whereas the program in the lecture notes uses 3 branches (BRP 05, BR 10, and BR 01). Assume that the following items in this order will be placed in the In-basket: 4, 10, 15, 20, and 25, one at a time. (The 4 is the count of numbers that follow.) What will the Out-basket contain after the program is executed? First try to understand each instruction thoroughly and then trace the execution of each instruction. Next write brief and precise comments which describe what each instruction does. Note that we initialized memory location 51 and 52 with 1 and 0, respectively. Memory location 52 will eventually store the sum (total) of four input values (10, 15, 20, and 25). Memory location 50 stores the current count, which is initially set to 4. Memory location 51 stores 1, a constant, by which memory location 50 will be decreased each time the loop is executed. Note that the program starts at address 00.

Address	Instruction (Mnemonics)	Comments start with // (for you to fill in)
00	IN	// Input 4
01	STO 50	// Store 4 in mailbox 50
02	BRZ 09	// If the calculator = 0 go to the instruction at address 09, else execute next instruction
03	IN	// Input 10, 15, 20, and then 25 one at a time
04	ADD 52	// Add number stored in mailbox 52 to the number in the in-basket. (0+10+15+20+25=70)
05	STO 52	// Sum is stored in mailbox 52 (eventually storing the total)
06	LDA 50	// Load the count in mailbox 50
07	SUB 51	// Subtract the constant, 1, from the count in mailbox 51
08	BR 01	// Brach to 01 to store the next count (4,3,2,1,0) in the mailbox
09	LDA 52	// Load the total number from mailbox 52, (70)
10	OUT	// Output 70
11	HLT	// coffee break

Address		Contents	Comments
50	DAT	?	// Stores the count 4 >> 0
51	DAT	1	// Stores constant of 1 to decrease the counter
52	DAT	0	// Stores the sums and eventually the total, 70

6. Assume now that the program from problem 5 will only read 3 numbers. That is, the following numbers in the following order will be placed, one at a time, in the In-basket: 2, 16, and 40, where 2 is the count of numbers that follow, and 16 and 40 are the numbers that are to be added. The first column in the table below 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, Out-basket, Accumulator, and Memory locations 50, 51, and 52 **after** each instruction is executed. Memory locations 51 and 52 are initialized with 1 and 0, respectively. The entry 00 → 01 in the PC column means that the PC is 00 when the instruction IN started and is changed to 01 when the instruction IN is finished.

The sequence in which instructions are executed	PC before → after	In-basket	Accumulator	Memory location 50	Memory location 51	Memory location 52	Out-basket
IN	00 → 01	2	2	?	1	0	?
STO 50	01 → 02	2	2	2	1	0	?
BRZ 09	02 → 03	2	2 **Branch not executed**	2	1	0	?
IN	03 → 04	16	16	2	1	0	?
ADD 52	04 → 05	16	16	2	1	0	?
STO 52	05 → 06	16	16	2	1	16	?
LDA 50	06 → 07	16	2	2	1	16	?
SUB 51	07 → 08	16	1	2	1	16	?
BR 01	**08 → 01**	16	1	2	1	16	?
STO 50	01 → 02	16	1	1	1	16	?
BRZ 09	02 → 03	16	1 **Branch not executed**	1	1	16	?
IN	03 → 04	40	40	1	1	16	?
ADD 52	04 → 05	40	56	1	1	16	?
STO 52	05 → 06	40	56	1	1	56	?
LDA 50	06 → 07	40	1	1	1	56	?
SUB 51	07 → 08	40	0	1	1	56	?
BR 01	**08 → 01**	40	0	1	1	56	?
STO 50	01 → 02	40	0	0	1	56	?
BRZ 09	**02 → 09**	40	0 **Branch is executed, go to address 09**	0	1	56	?
LDA 52	09 → 10	40	56	0	1	56	?
OUT	10 → 11	40	56	0	1	56	56
HLT	11 → 11 or 11 → 00	19 or 0	56 or 0	0 or 0	1 or 0	56 or 0	56 or 0