Asn 7

1) X+Y-Z

2) Assembly Language

Label	Instruction
	Lda X
	Sub Y
	Add W
	Sub Z
	Out
	Hlt
X:	Dat -1
Y:	Dat 8
W:	Dat 3
Z:	Dat -113

Symbol Table

Label	Addr
X:	12
Y:	13
W:	14
Z:	15

Calculates: 107 or 0x6B

3) Machine Code

	Loc	Hex	Bin
Lda X	0	0C	0000 1100
Sub Y	1	1D	0010 1101
Add W	2	2E	0001 1110
Sub Z	3	3F	0010 1111
Out	4	E0	1110 0000
Hlt	5	F0	1111 0000

4) 2^{16} more bits

5)

Label	Instruction
	Lda x
	Out
Top:	Add y
	Out
	Jmp Top
x:	Dat 0
y:	Dat 1

Symbol Table

Label	Address
x:	14
y:	15

Output! 0, 1, 2, 3, 4....

6)

Label	Instruction
	Lda X
	Add X
	Sub Y
	Sta X
	Hlt
X:	Dat 75
Y:	Dat 120

Symbol Table

Label	Address
X:	14
Y:	15

Should Compute:
75+75 = 150
150-120 = 30

Actually computes!
75+75 = 150-256 = -106
-106-120 = -226+256 = 30

The pitfall is that the program computed the correct value but only because the second overflow corrected it.

9)

Current			Next machine state			
##	PC	instr	PC	ACC	out	hex mem contents
0			0	–	–	0E 1E 2F 3E Fx xx xx ... 4B 78
1	0	Lda	1	4B	–	same
2	1	Add	2	96	–	same
3	2	Sub	3	1E	–	same
4	3	Sta	4	1E	–	0E 1E 2F 3E Fx xx xx ... 1E 78
5	4	Hlt	xx	1E	–	same

10)

Label	Instruction
	Lda X
Top:	Out
	Sub Y
	Jaz Done
	Jump Top
Done:	Out
	Hlt
X:	Dat 55
Y:	Dat 5

Symbol Table

Label	Address
X:	14
Y:	15

11) Just change X to X: Dat 10 to modify the program to output 10, 5, 0.

Current ##	PC	Instr	Next Machine State PC	Acc	Out	hex mem contents
0			0	—	—	0E EX 2F 55 41 EX Fx xx...0A 05
1	0	Lda	1	0A	—	same
2	1	Out	2	0A	0A	
3	2	Sub	3	05	—	
4	3	Jaz	4	05	—	
5	4	Jump	5	05	—	
6	5	Out	6	05	05	
7	6	Sub	7	00	—	
8	7	Jaz	8	00	—	
9	8	Out	9	00	00	
10	9	Hlt	xx	00	—	