4. Below is a Little Man program that implements exercise 6.9, p. 194. The program is very similar to the last LMC program in the posted lecture notes on chapter 6. 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, 34, 17, 19, and 20, 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 the brief and precise comments that describe what each instruction does. Note that we initialized memory location 81 and 89 with 1 and 0, respectively. Memory location 89 will eventually store the sum (total) of four input values (34, 17, 19, and 20).

Address	Instruction (Mnemonics)	Instruction code	Comments start with // (for you to fill in)
00	IN	901	// stores the contents of in-basket in the calculator (4)
01	STO 80	380	// stores the contents of the calculator (4) in mem location 80
02	BRZ 09	709	// go to address 09, if calc is 0. since calc is 4 it will instead go to the next instruction.
03	IN	901	// stores contents of in basket in the calc (34)
04	ADD 89	189	// adds the contents of calc + the value stored at memory location 89, which is 0.
05	STO 89	389	// The contents of the calculator are stored at memory location 89, which is 34.
06	LDA 80	580	// loads the contents of mem location 80 which is 4.
07	SUB 81	281	// subtracts the contents of memory location 81, which is 1, from the calculator. This leaves 3 in the calc.
08	BR 01	601	// Unconditional branch brings us to address 01, which tells us to store the contents of the calc (3) in address 80. The process repeats until all numbers have been read in.
09	LDA 89	589	// Loads the contents of mem location 89 to the calc (90
10	OUT	902	// Writes result: 90
11	HLT	000	// coffee break

Address		Contents	Comments
80	DAT	?	// unknown at first stores count
81	DAT	1	// used to decrement count by 1
...			
89	DAT	0	// stores sum.

} data area.

5. Assume now that the program from problem 4 will read only 3 numbers. That is, the following numbers will placed, one at a time, in the In-basket: 2, 15, and 34, where 2 is the count of numbers that follow, and 15 and 34 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 80, 81, and 89 **after** each instruction is executed. Memory locations 81 and 89 are initialized with 1 and 0, respectively. The entry $0 \rightarrow 1$ in the PC column means that the PC is 0 when the instruction IN started and is changed to 1 when the instruction IN is finished.

The sequence in which instructions are executed	PC before → after	In-basket	Out-basket	Accumulator	Memory location 80	Memory location 81	Memory location 89
IN	$0 \rightarrow 1$	2	?	2	?	1	0
STO 80	1→2	2	?	2	2	1	0
BRZ 09	2→3	2	?	2	2	1	0
IN	3→4	15	?	15	2	1	0
ADD 89	4→5	15	?	15	2	1	0
STO 89	5→6	15	?	15	2	1	15
LDA 80	6→7	15	?	2	2	1	15
SUB 81	7→8	15	?	1	2	1	15
BR 01	8→1	15	?	1	2	1	15
STO 80	1→2	15	?	1	1	1	15
BRZ 09	2→3	15	?	1	1	1	15
IN	3→4	34	?	34	1	1	15
ADD 89	4→5	34	?	49	1	1	15
STO 89	5→6	34	?	49	1	1	49
LDA 80	6→7	34	?	1	1	1	49
SUB 81	7→8	34	?	0	1	1	49
BR 01	01→02	34	?	0	1	1	49
STO 80	02→03	34	?	0	0	1	49
BRZ 09	03→04	34	?	0	0	1	49
LDA 89	04→05	34	?	49	0	1	49
OUT	05→06	34	49	49	0	1	49
HLT	0	0	0	0	0	0	0