

illustrates another feature of the CODE command:

```
100 MACHINE 200
110 CODE"LDA,FF,9F,CMPIM,34,BEQ,@120,LDAIM,34,STA,FF,9F"
120 CODE"RTS"
200 END
```

At line 110 the number at memory location 9FFF is loaded into the accumulator. The CMPIM is the assembly language mnemonic for comparing the accumulator with the following number (34). BEQ is the code for "branch if equal". @ followed by a linenumber gives the location to go to if the accumulator equals 34. The program will check what is in location 9FFF. If it is 34, it will branch to line 120 and execute the RTS. If it is not, it will load 34 into the accumulator, store the 34 into location 9FFF, and execute the RTS. Whenever you use a branch command (e.g., BNE,BCC), it should be followed by a comma and then @ followed by a linenumber. Here is another example:

```
100 MACHINE 200
110 CODE"LDA,FF,9F,CMPIM,34,BEQ,@120,LDAIM,34,JMP,#130"
120 CODE"LDAIM,33"
130 CODE"STA,FF,9F,RTS"
200 END
```

The above program is similar to the previous one. First it compares the number in 9FFF with 34. If they are equal, it goes to line 120 and loads 33 to the accumulator, executes the code on line 130 which stores the 33 to 9FFF, and finally returns. If they are not equal, it loads 34 to the accumulator and executes the JMP command. This is the assembler mnemonic for JUMP, and the next two bytes must specify where it is to jump. Normally you follow JMP, JSR, and JMPI with comma, # sign, linenumber. The compiler inserts two bytes which represent the address of the first command on the given line. Thus, after loading 34, it jumps to the STA,FF,9F on line 130, which stores the 34 to 9FFF, and executes the RTS command.

Warning: The value of the X register must not be changed by the routine. If you must use the X register, it must be saved and then restored before the final RTS. TXA,PHA saves the X register and PLA,TAX restores it.

CODEL

The CODEL command is normally used to specify the address of a variable. When used in this way, there are several possible formats:

CODEL(variablename)

CODEL(variablename+integernumber)

CODEL(variablename+"L")

CODEL(variablename+"H")

Integernumber is a positive integer without a % sign at the end. You can also use CODEL to specify the address of a line number. (See the reference manual for information on this option.) The following program inputs a number, adds one to it, and prints the result: