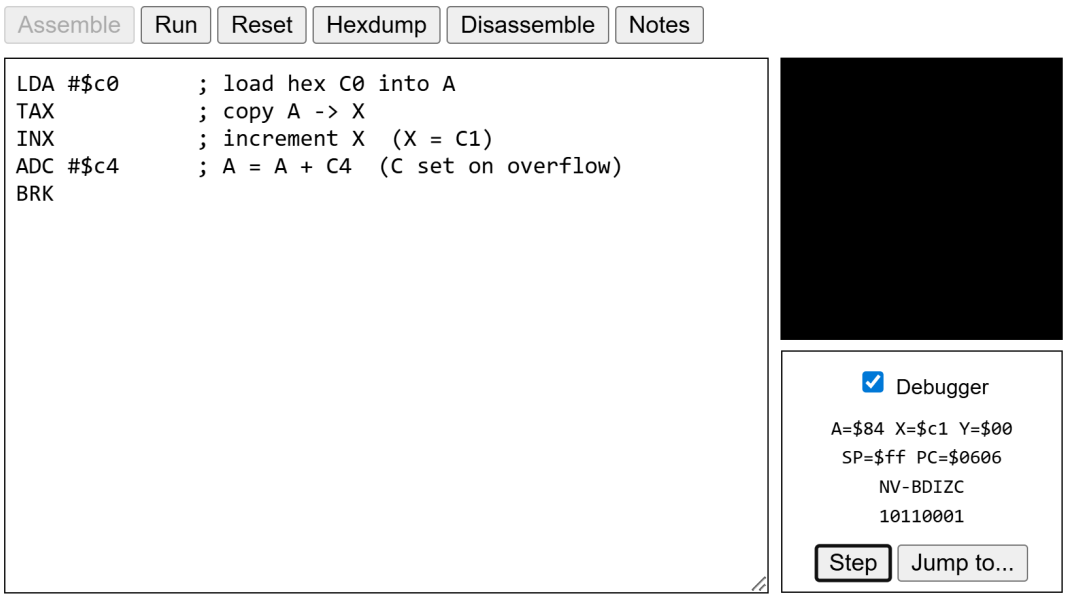
Donaven Bruce
CS130
October 13, 2025

Lab Exercise 3: Instructions

Part 1

I used the top editor in easy6502. I assembled the starter program, turned on the debugger, and stepped until after the ADC instruction. I was instructed to skip the Companion Example in Part 1.

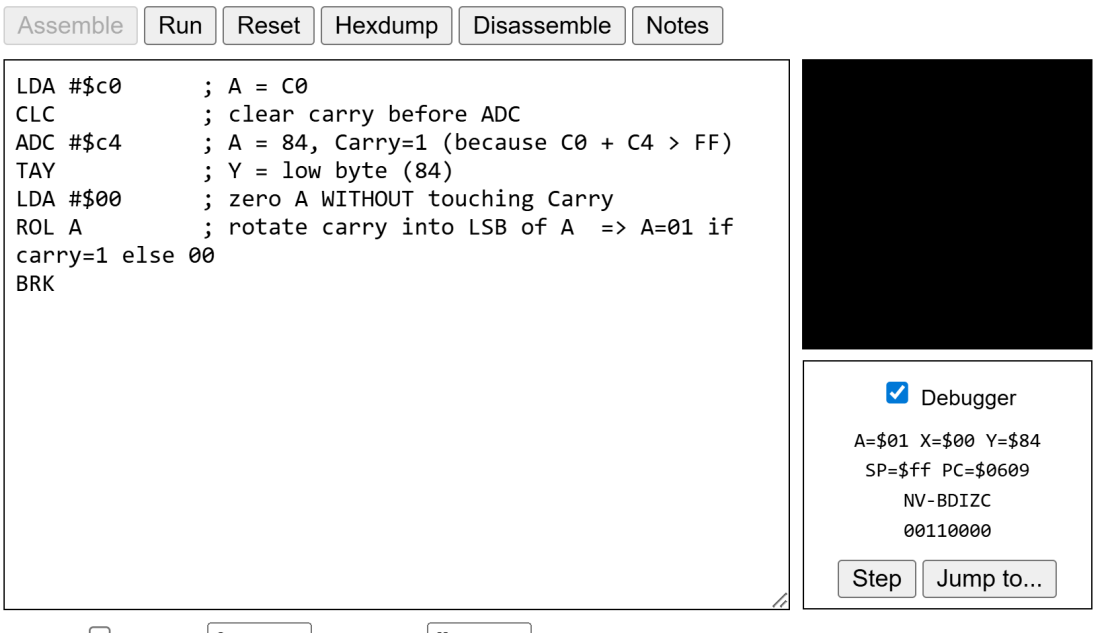
Step 6 result



Part 1 — Step 6: Post-ADC state showing A = $84 and C = 1.

My variation

I changed the program so the low byte of the sum is in register Y and the carry is in register A. I assembled and stepped to confirm the result.

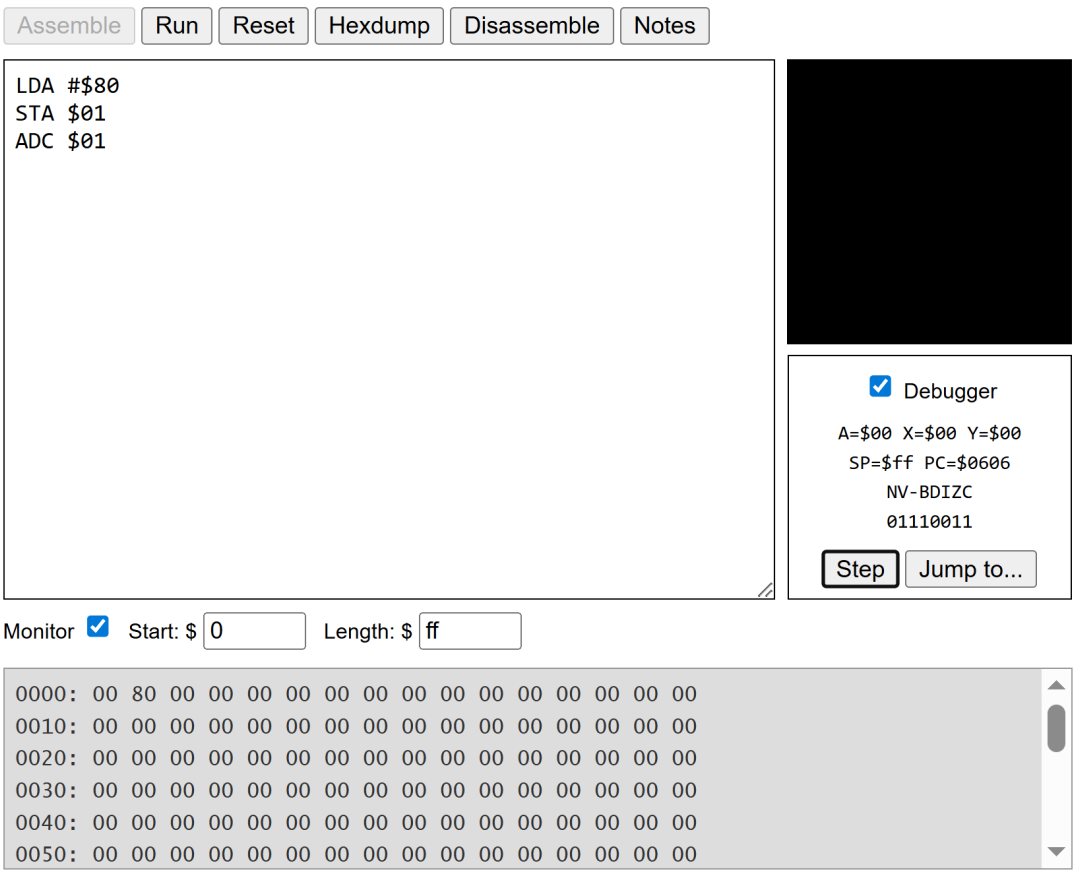


Part 1 — Variation: Y holds the low byte and A reflects the carry (A = $01).

Part 2

For Part 2 I used the lower editor. I turned on Monitor and Debugger, assembled, and stepped as needed. I captured each result as a screenshot.

Steps 7 to 9



Part 2 — Steps 7 to 9: After `ADC $01`. A = $00, C = 1, Z = 1.

Another example with FF

Displaying the correct sum

Assemble | Run | Reset | Hexdump | Disassemble | Notes

```
LDA #$ff
STA $01
ADC $01
```

Debugger

```
A=$fe X=$00 Y=$00
SP=$ff PC=$0606
NV-BDIZC
10110001
```

Step | Jump to...

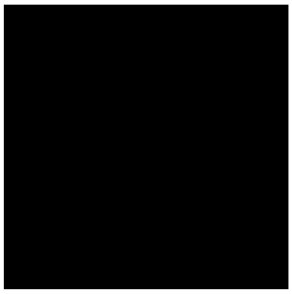
Monitor Start: $ 0 Length: $ ff

```
00a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ec
```

Part 2 — Another example with $FF: After `ADC $01`. A = $FE, C = 1, N = 1.

Assemble | Run | Reset | Hexdump | Disassemble | Notes

```
LDA #$85
CLC
ADC #$85
TAY
LDA #$00
ROL A
TAX
BRK
```



Debugger

```
A=$01 X=$01 Y=$0a
SP=$ff PC=$060a
NV-BDIZC
01110000
```

Step | Jump to...

Monitor Start: $ 0 Length: $ ff

```
00a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 27
```

Part 2 — Displaying the correct sum for $85 + $85: Y = $0A, A = $01, X = $01.

Step 10 answers

Q1 Can I run it after removing the two $ signs?
A No. It does not assemble.

Q2 What happened?
A The assembler rejects the values without the $ prefix.

Q3 Conclusion
A The $ prefix is required for hexadecimal. Without it the values must be decimal.

Q4 Values needed to assemble and reproduce the original results
A Use decimal 192 and 196.

Step 11

I created a branching version that uses register Y as the counter. I will submit this program as a separate text file named Bruce Lab 3 Step11.txt.