

<i>Source Code</i>	<i>Machine Code</i>
<pre> {     int a     a = 1     while (a != 5) {         a = 1 + a         print(a)     } } \$ </pre>	<pre> A9 00 8D 52 00 A9 01 8D 52 00 AD 52 00 8D 54 00 A9 05 8D 53 00 AE 54 00 EC 53 00 A9 00 D0 02 A9 01 A2 00 8D 53 00 EC 53 00 D0 20 AD 52 00 8D 53 00 A9 01 6D 53 00 8D 52 00 AC 52 00 A2 01 FF A9 00 8D 53 00 A2 01 EC 53 00 D0 BF 00 </pre>

### *6502a Machine Code Disassembly*

0000	A9 00	LDA #\$00	<b>int a</b> (0x52)
0002	8D 52 00	STA \$0052	
0005	A9 01	LDA #\$01	<b>a = 1</b>
0007	8D 52 00	STA \$0052	
000A	AD 52 00	LDA \$0052	<b>while (a != 5) {</b> Copy a to t <sub>2</sub> (0x54). Copy the compare-to value to t <sub>1</sub> (0x53). Compare t <sub>2</sub> and t <sub>1</sub> , and assign Z flag. Acc = 0. If t <sub>2</sub> != t <sub>1</sub> branch to 0x21. (if t <sub>2</sub> == t <sub>1</sub> ) Acc = 1. X register = 0. Store Acc in t <sub>1</sub> (0x53). Compare t <sub>1</sub> and X reg, and branch if unequal.
000D	8D 54 00	STA \$0054	
0010	A9 05	LDA #\$05	
0012	8D 53 00	STA \$0053	
0015	AE 54 00	LDX \$0054	
0018	EC 53 00	CPX \$0053	
001B	A9 00	LDA #\$00	
001D	D0 02	BNE \$0021	
001F	A9 01	LDA #\$01	
0021	A2 00	LDX #\$00	
0023	8D 53 00	STA \$0053	
0026	EC 53 00	CPX \$0053	
0029	D0 20	BNE \$004B	

002B	AD 52 00	LDA \$0052	
002E	8D 53 00	STA \$0053	
0031	A9 01	LDA #\$01	<b>a = 1 + a</b>
0033	6D 53 00	ADC \$0053	
0036	8D 52 00	STA \$0052	
0039	AC 52 00	LDY \$0052	
003C	A2 01	LDX #\$01	<b>print(a)</b>
003E	FF	SYS	
003F	A9 00	LDA #\$00	}
0041	8D 53 00	STA \$0053	unconditional jump to the
0044	A2 01	LDX #\$01	top of the loop, \$000A,
0046	EC 53 00	CPX \$0053	which is 0xBF (191) bytes
0049	D0 BF	BNE \$000A	forward, which is actually
004B	00	BRK	64 bytes backward.
			break