Eric Roberts
MLA 321

Handout #6
January 26, 2016

Assignment #4

Due: Tuesday, February 2

Write out answers to each of these problems. Screenshots from the Toddler application would be the best way to show your work.

Problem 1—The Toddler machine

Answer all three parts of the exercises in the puzzle box on page 120.

Problem 2—Assembly language

Translate the following assembly language program into Toddler machine instructions, showing the contents of all memory addresses loaded by the program:

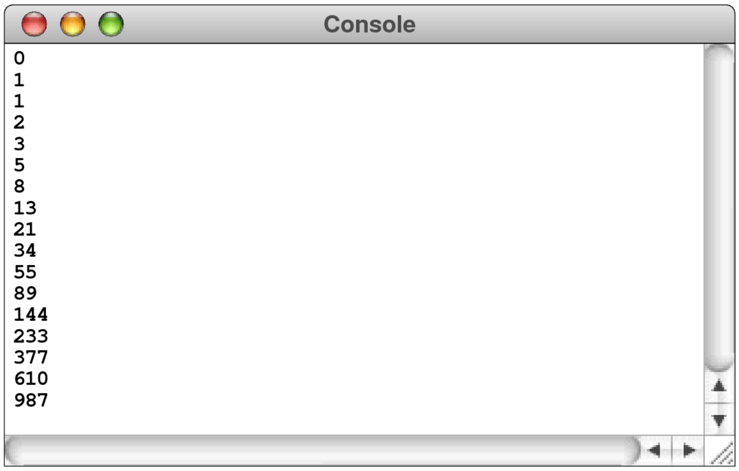
```
start:   INPUT   n
         LOAD    #0
         STORE   total
         LOAD    #1
loop:    STORE   i
         LOAD    n
         SUB     i
         JUMPN   done
         LOAD    total
         ADD     i
         ADD     i
         SUB     #1
         STORE   total
         LOAD    i
         ADD     #1
         JUMP    loop
done:    OUTPUT  total
         HALT

i:       0
n:       0
total:   0
```

You can, of course, simply type this program into the simulator and copy down the machine instructions, but you will understand things better if you do the translation by hand and then use the simulator to check your answer.

What output does this program produce if you run it and enter 5 as the input value in response to the first instruction? What value does this program compute in general?

Problem 3—Fibonacci numbers

Write a Toddler program to generate the Fibonacci numbers that you worked with on Assignment #2. Your program should run until the values stop increasing, which happens when the answer no longer fits in a three-digit word. The output should therefore look like this:



```
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
```

The next term would be 1597 (610 + 987), but Toddler would store only the last three digits. Since 597 is smaller than 987, your program should stop.

Problem 4—The Von Neumann architecture

Consider the following machine-language program for the Toddler machine:

```
(01)  +104
(02)  +402
(03)  +205
(04)  +905
(05)  +500
```

If you look at the operation codes, you can see that this program consists of a **LOAD**, **SUB**, **STORE**, **OUTPUT**, and **HALT** instruction, in that order. As a result, you might expect it to subtract one number from another, output the result, and halt. It doesn't. What does this program do when you run it on the Toddler simulator? In a sentence or two, explain what is going on in this program that causes it to violate your intuition.