

Assignment #4

Due: Thursday, October 27

Send me email with 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 130.

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

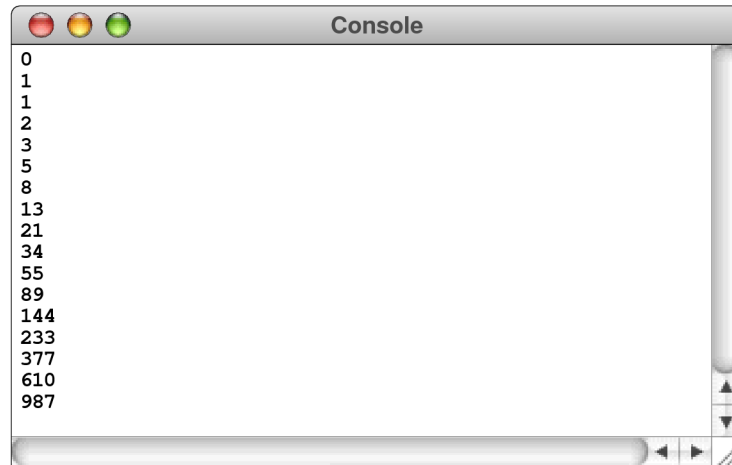
i:      0
n:      0
total:  0
```

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 solved for the Analytical Engine in 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:



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—Greatest common divisor

Write a function `mod` for the Toddler machine that takes the values of two variables, `num` and `den`, and then uses repeated subtraction to compute the remainder of `num` divided by `den`, which should be returned in the `AC`. Use your `mod` function to implement a second function `gcd` that takes its inputs from the variables `x` and `y` and then uses Euclid's algorithm to compute the greatest common divisor, which is again returned in the `AC`. Once you have written these two functions, you should be able to use the following main program to compute the greatest common divisor of any two positive integers:

```
INPUT x
INPUT y
CALL gcd
STORE result
OUTPUT result
HALT
```

Your program will need to declare variables for `x`, `y`, `num`, `den`, and `result`, as well as supplying the code for `mod` and `gcd`.

Problem 5—Pedagogy and the Toddler machine

Identify some aspect of the Toddler machine that you found confusing in either the text or the classroom presentation. Write up a short paragraph describing the source of your confusion and then submit one or two slides that illustrate the relevant concept.