Name: Sang Shin
Date: 01/15/2014

Milestone Report

Specification (what do <u>you</u> think the purpose of this milestone is)

I think the purpose of this milestone is to get us to understand the programming style of gforth. Additionally, it helps us to think in postorder since the syntactic format for gforth is postorder.

Processing (how did <u>you and/or your team</u> go about solving the problem)

I solved the majority of the problems by converting the infix style expressions in the exercises to expression trees. From this, I performed a postorder traversal and it was simple to generate the gforth input. Additionally, reading the gforth manual helped with syntax and understanding floating point.

Testing Requirement (how did <u>you and/or your team</u> test for correctness)

Testing the mathematical expressions were simple. check to see if the gforth code outputted correct answers and compare them to the actual answers to the mathematical expressions. As for testing the functions for "converint," "fac," and "fib" general test cases were used to see if the function outputted the proper answer. For "convertint," I used integer values to determine if the output became floating point. For "fac," I used 0 to check for the case ~~cat~~ statement and 5 and 10 to check if the function worked. For "fib," I used two tests. The first test was to check if giving an input n to the fib function, the appropriate fib number is outputted. The second test was to check if the fib numbers work with a loop. An off-side test of fib was to input a negative number.

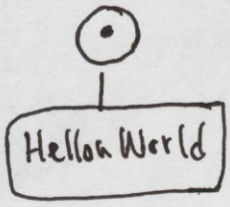
Retrospective (what did <u>you</u> learn in this milestone)

I have learned a lot in this milestone. The first is the programming style and language of gforth. The second is understanding the postorder traveral of an expression tree to generate gforth input. The third is understanding how to compile the gforth code.

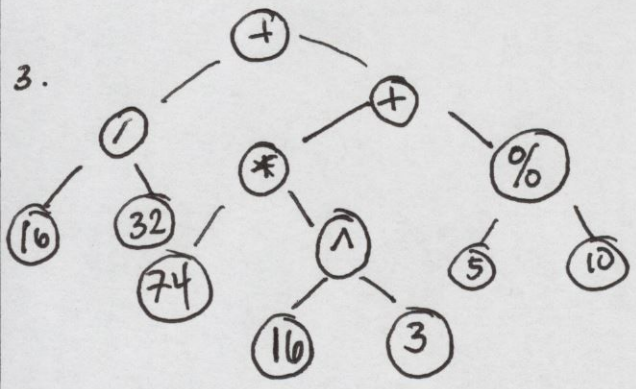
Team Evaluation (what is the percentage of time contributed by each team member)

Sang Shin – 100%

1.

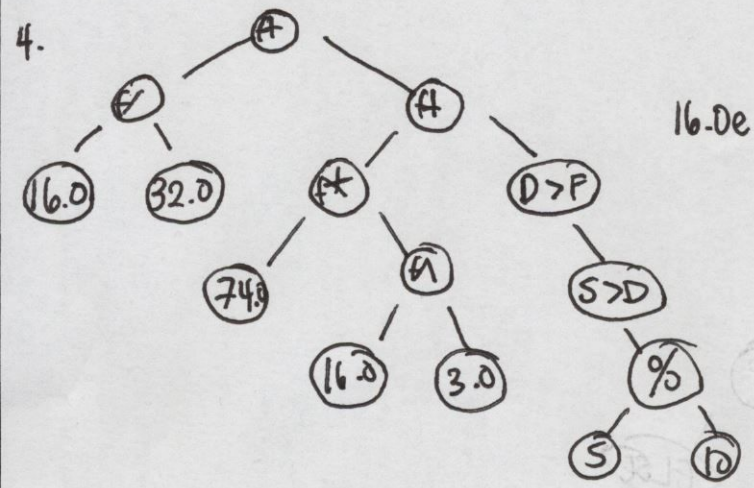


3.



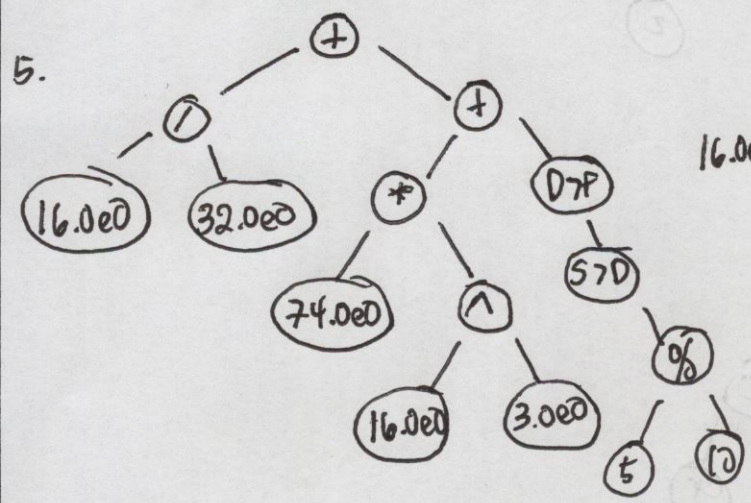
16 32 / 74 16 3 ^ * 5 10 % ++

4.



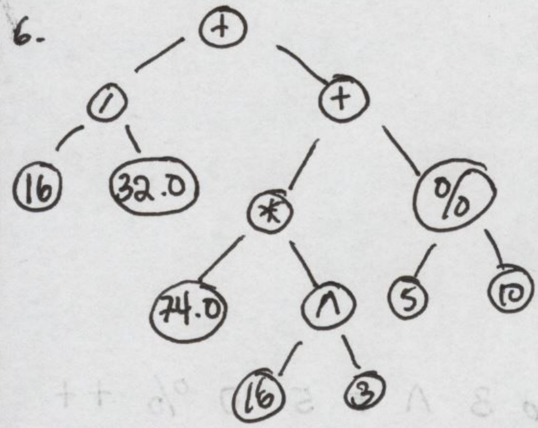
16.0e 32.0e f/ 74.0e 16.0e 3.0e ^ f* 5 10 % f+f

5.

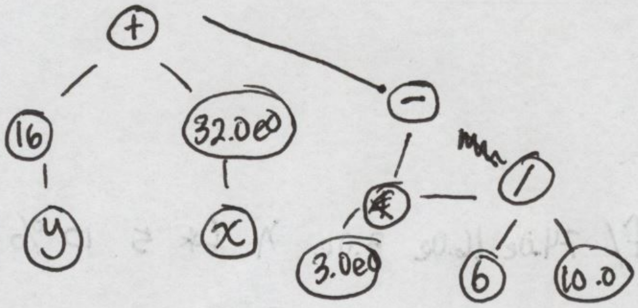


16.0e0 32.0e0 F/ 74.0e0 16.0e0 fdup fdup F* F* F*
5 10 mod S>D D>F F+ F+

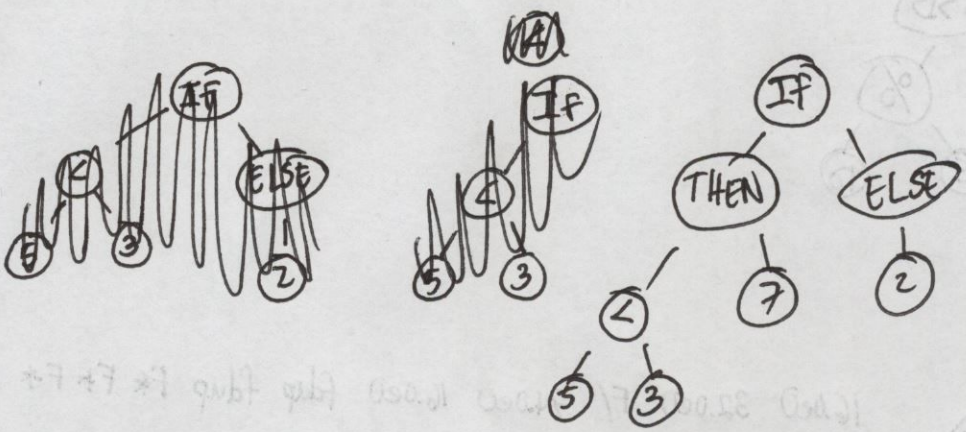
6.


+
/
+
16
32.0
*
%
74.0
^
5
10
16
3

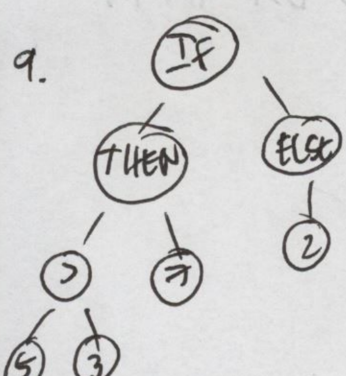
7.


+
16
32.0e0
−
y
x
*
/
3.0e0
6
10.0

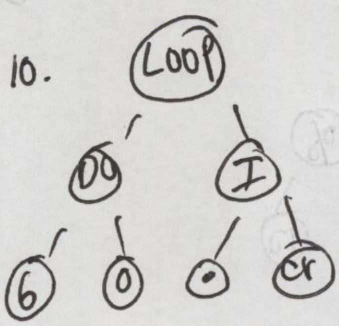
8.


IF
THEN
ELSE
<
7
2
5
3

9.


IF
THEN
ELSE
>
7
2
5
3

10.


LOOP
DO
I
6
0
.
cr