

# EE4371: Assignment 2

August 18, 2019

1. Read Chapter 2 of Tanenbaum
2. Read Chapter 2 of Aho, Hopcroft and Ullman
3. Implement a program to evaluate a postfix expression. A postfix expression input has the following form:

tag tag tag ... tag

where each tag can be one of

- (a) A string representing a number
- (b) An operation (one of +, -, \*, and /). Each operation takes two arguments, and pops and acts on the latest two values on the stack. The answer is pushed back on the stack.

Tags are separated by a combination of blanks or tabs. The '\n' character terminates the expression. An example of a valid postfix expression is

1.5 2.8e3 -12 + \*

Also note that both integer and real numbers are to be accepted.

**The program should use array based stacks and should call functions to do all operations.** It should extract tags, validate them, store them in a stack if numbers and operate on them using the operators. The output should be the value of the expression.

The code should be able to handle upto 5 values.

If the expression is invalid, an output should come that states this and indicates the why the expression has been rejected.

Egs:

1.5 2 -3 + *	-1.5
1.5 2 + -3 *	-10.5
1.5 2 + * -3	Invalid expr. "*" does not have enough operands
1.5 2e2 + -3e2.5 *	Invalid expr. Illegal tag: -3e2.5

4. Repeat problem 3 using a linked list to implement the stack. You should need to change the functions push and pop (and add initial code to create the linked list), but main code should not change.

**Note:** I will test your code with varied inputs. Please ensure you can handle all corner cases.

Submit your program to moodle. Comments are required, not optional. However, no report is to be submitted.