

Upload your solution (as a zip file) to the ODTUClass site for the course. The **deadline is Jan. 20, 2019**

1. HW2

Write a program that will read an input file (e.g. `input.txt`) line by line and perform the calculation on each line. The results will be written back to another file (e.g. `output.txt`). Each line of the input file will contain `+`, `-`, `*`, `/` and floating point numbers separated by spaces. An example input file (`HW2_test/input.txt`) is given with this question.

To solve the problem, read the article about the Shunting-yard Algorithm and RPN, carefully:

https://en.wikipedia.org/wiki/Shunting-yard_algorithm

https://en.wikipedia.org/wiki/Reverse_Polish_notation#Postfix_algorithm

You will implement these algorithms in order to solve the problem

You may re-use the queue and stack implementations we prepared in class:

<https://github.com/acacar/MIN545/tree/master/Lecture08/MyCollections>

You can test your program with the `HW2_test/input.txt` given with this homework. Your output should match `HW2_test/correct_output.txt`

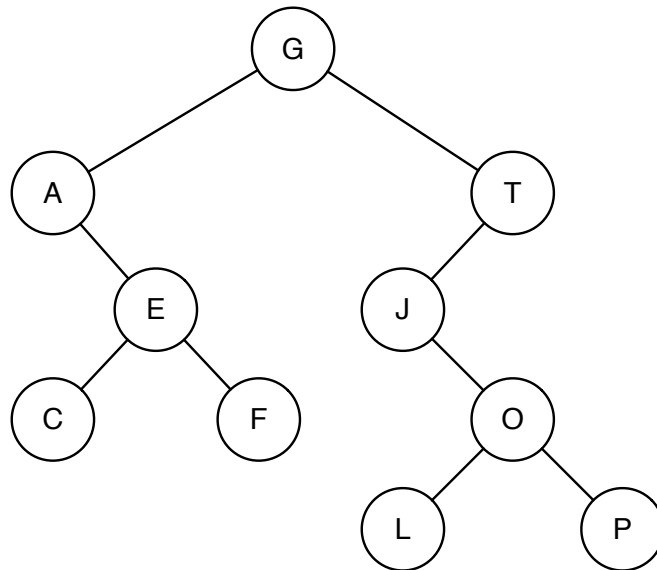
BONUS: If you can make your program handle parentheses correctly as well, you will get a 10% bonus. The input file will then contain `(`'s and `)`'s as well. There is a sample `HW2_test/bonus_input.txt` given with this file. If you do the bonus part, your program's output should match `HW2_test/correct_bonus_output.txt`, which is also given.

2. HW3

You are to write a small Python program that, when given the post-order traversal of a **binary search tree**, will reconstruct the tree and return its pre-order traversal. The tree nodes will each hold one uppercase letter. For example, let's say the post order traversal of such a binary search tree is:

CFEALPOJTG

The only binary search tree with this traversal is:



Therefore, the pre-order traversal would be:

GAECFTJOLP

So, a sample run of the requested program for the above example would then be:

Enter post-order traversal: CFEALPOJTG

The pre-order traversal is GAECFTJOLP

The strings given to your program will not consist of any repeating characters, so you are guaranteed that the longest string will only have 26 characters in it.