Instructions for the Expression Tree Homework

Input is RPN. Read in each line of input and generate an expression tree.

To make it simpler, you may assume that the value of all variables is ***zero*** when evaluating the expression. (It would be nicer if you put in values of variables to see if it works +20 for doing so.)

Expressions consist of constants (which are double precision numbers)

You must implement all the following optimization rules. For any expression  
x \* 0 → 0

x \* 1 → x

x + 0 → x

x - x → 0

| RPN | Tree | Optimization (if blank, then same as the tree) | eval() All variables will be set to zero for this homework for eval. |
| --- | --- | --- | --- |
| 3.5 4 + |  |  | 7.5 |
| x 1 + |  |  | 1 |
| x 0 + |  |  | 0 |
| a b + |  |  | 0 |
| x 1 + 0 \* |  |  | 0 |
| 4 sqrt |  |  | 2 |
| x 2 + sqrt |  |  | 1.414.... |
| x 1 + neg |  |  | -1 |

The following optimizations are not required, but if you do them you will receive an additional 100 points

| RPN | Tree | Optimization (if blank, then same as the tree) | eval() All variables will be set to zero for this homework for eval. |
| --- | --- | --- | --- |
| a 3 \* a 4 \* + |  |  | 0 |
| a 2 + a 3 \* + |  |  | 2 |
| x x + x - |  |  | 0 |
| x x + x 1 + - |  |  | -1 |