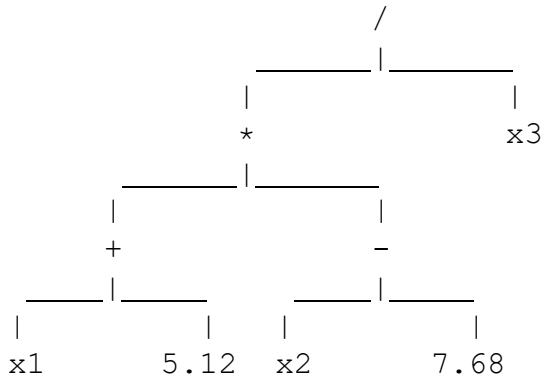
Write a C program that takes as input a fully parenthesized, arithmetic expression of binary operators $+, -, *, /$, and converts the expression into a binary expression tree. Your program should take input from the command line. The entire expression should be in a character string without any space in it.

An input string only includes floating numbers in the format of Y.YY, that is, one digit to the left of the decimal point and two digits to the right of the decimal point, and variables of the form of x1, x2, ...

Your program shall allow for the leaves in the expression tree not only to store floating values but also to store variables of the form x1, x2, x3, ..., which are initially 0.0 and can be updated interactively by the user. For example, expression $(((x1 + 5.12) * (x2 - 7.68))/x3)$ will be converted into a binary expression tree like:



Your program should then show a menu with the following options:

1. Display
2. Preorder
3. Inorder
4. Postorder
5. Update
6. Calculate
7. Exit

Description:

(a) When option 1 is selected, your program should display the tree in some way so that the tree can be visualized, and also print the name and value of each variable, like "`x1:0.0`" when `x1` initially has value 0.0.

(b) If an option of 2, 3 or 4 is selected, your program should print the expression by the corresponding traversal order (Note: no parentheses for preorder and postorder traversal but fully parenthesized for inorder traversal).

(c) Option 5 requires further input from user. A pair of input, namely,

variable_name, new_value

will be provided interactively and your program should search for the *variable_name* and replace its value by *new_value*.

(d) Arithmetic calculation is invoked by option 6 which shall display the calculation result. Your program should be able to detect an error of *divide-by-zero* before calculation.

(e) Option 7 terminates your program.