Using the expression tree, you need to first start by adding the first set of parentheses around the entire equation, then you start walking through the equation, the stack will be a parenthesis and then it will have a number, once it sees the next operator it will make the number the operator’s left child and make the number after the operator the right child. Once it reads in the close parenthesis it will pop the open parenthesis out of the stack so that the only thing in the stack will be the first operator. Then it will see the next operator, likely the root operator and make the first operator the left child of that operator. Then it will see the second set of parenthesis, and make a new stack node and and read in the number of the second child operation, then it will read in the third operator which will make the number before it the left child and the next number the right child. This leaf of the operator will then be made the left child of the root operator.

For printing the prefix, it will walk through, print first the root operator then go to the right child of the root operator and put the operator before the left child number, then the left child number, then the right child number, it will add the parenthesis around this child of the root operator. Finally it will add the last set of parenthesis and walk through the right child of the root operator and print out the operator of the right child, then it will print out the left child of the right child operator then the right child operator’s right child number. It will add the remaining parenthesis and print it out.

The same thing will be done for the postfix except it will do the left child first then the right child then the operators for each of the sub equations then it will do the root operator then add the parenthesis.