**Project Description**

This program shows displays a truth table for a given proposition, i.e. p & q. I got the idea from the beginning of discrete math course when we discussed logic, logical operators, prepositions and truth table. The program goes through following steps after a data input:

1. It checks for errors, if it finds any, it will report the user and asks for the input again
2. It converts the preposition (the input) to postfix expression. For example, the postfix form of 3 + 4 is 3 4 +.
3. It makes a simple binary tree filling the nodes with operators and the leaves with operands. It makes this binary tree for each column of the truth table except the columns with a single operand. For example, if the proposition is something like ( p & q ) | q, it will make a tree that represents p & q, and ( p & q ) | q.
4. It gets the list of operands in the given proposition and sorts them alphabetically
5. It generates initial truth table values for the single operands only. For example, if the operands are p and q, it generates T, T, F, F and T, F, T, F for p and q column respectively.
6. It passes that initial truth table to the evaluator
7. Then, for each tree, it evaluates the tree using the evaluator and saves the result in the form of a list.
8. Finally, it will display the result in a table form.

I chose to do this project because it involves the usage of binary data structure as binary expression tree. I thought it would be fun yet challenging and improve my programming skills with binary trees.