

/*-----DESIGN EXPLANATION Q1: Stack Postfix-----*/

This program is made with 4 major functions that performs Postfix. Main function creates two files for input and output and uses its values to perform postfix. The functions evaluate operands and creates stack to store the data and performs calculations: addition, subtraction, multiplication, division.

linkedStackType:

In this program, we aim to build stacks using linked list.

To do that, the all functions need to be able to test and perform that is needed in stack.

The most important functions are evaluateExpression and evaluateOpr.

evaluateExpression tests if the passed in operand are #, if it is, it saves on the file and if they are not it performs evaluateOpr. evaluateOpr test the passed in operands and if they are +, -, *, /, it creates a stack to store numbers and uses top two numbers to perform the calculation with passed in operand. After the calculation, the operands and numbers and popped out and only the result is saved in the stack. When there's only one value left as the result, that the end of the calculation.

Method:

```
void evaluateExpression(ifstream& inpF, ofstream& outF, stackType<double>& stack, char& ch, bool& isExpOk)
```

```
void evaluateOpr(ofstream& out, stackType<double>& stack, char& ch, bool& isExpOk)
```

```
void discardExp(ifstream& in, ofstream& out, char& ch)void
```

```
void printResult(ofstream& outF, stackType<double>& stack, bool isExpOk)
```