**CSCI 301 Moustafa Elsayed**

**Computer Science 2**

**Project 5: Evaluating Postfix expressions using a stack**

**Introduction**

A stack is helpful in quickly accessing data that is recently added to a list. This program uses a stack to hold the integers and signs that are written in an input file. Then the program solves the expression that is written in Postfix notation and prints the result in the terminal.

**Data Structures**

This program uses 9 data structures:

* A char type “data”: In the header and implementation files, this is used to hold the items in the input file and is placed in the stack.
* A stack: This is used to hold the items in the file and order to be the first item added is the last item to be removed.
* An int type “used”: In the header and implementation files, this is used to count the number of items in the stack.
* An ifstream type “inFile”: In the main file, this is used to hold the input file.
* An int type “result”: In the main file, this is used to hold the result of the arithmetic expression.
* A char type “x”: In the main file, this is used to hold each item in the input file.
* Int types “opnd1” and “opnd2”: In the main file, these are used to hold the last two integer added to the stack.
* A string type “name”: In the main file, this is used to hold the name of the input file.

**Functions**

This program uses 3 functions:

* pop(): In the header and implementation and main files, this function is used to return the last value added to the stack and decrement the value of “used”.
* push(): In the header and implementation and main files, this function is used to add an integer to the stack.
* apply(): In the main file, this program is used to apply the arithmetic expression based on what is entered in the parameters of this function.

**The main program**

In the main program the user is asked for the file name, then the input file is opened if the correct input file name is written. The program then prints “The expression is:” to the user and opens a loop. At the beginning of the loop the program takes the first character in the file from the left then prints that character to the user, then the program checks if the character is an arithmetic sign or not, if it is an arithmetic sign then the program uses the “pop()” function twice to hold the last two items added to the stack to the variables “opnd1” and “opnd2”, then the program calls the function “apply()” and returns to the variable “result”, and then we call the function “push()” to add the “result” variable to the stack. If the character is a number, then the program calls the function “push()” to add the number to the stack. After the loop ends the program prints the result.