**WEEK 6**

**AIM:** Evaluate the arithmetic expression using stack

In this exercise, I am reading postfix expressions from the input file, evaluating the expression, and storing the final result in the output file.

**VARIABLES USED**

Postfix expression: exp[100]

Popped elements: op1, op1

Stack for storing operands: stack[50]

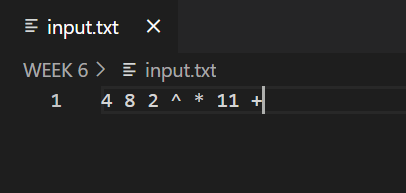
Top of the stack: top

Element to push: elem

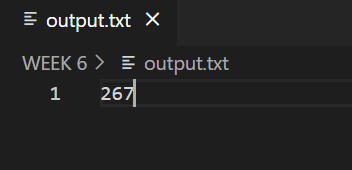
File pointer: fptr

Helper variables: flag, i, ch

Sample Input: 4 8 2 ^ \* 11 +



Sample Output: Evaluated answer: 267



Algorithm:

Step1. Read a word(seperated with a whitespace) from postfix expression

Step2. If the word is a digit, then push it onto the stack

Step3. If the character is an operator, pop two operands from the stack.

Step4. After popping two operands, apply the operation on both the operands and push the result onto the stack.

Step5. Repeat step1 to step4 till null character reached (postfix expression is completely processed).

Step6. Print the top of the stack.

Space Complexity: O(n)

Time complexity: O(n) // n is the number of characters in expresion