Nazara Ibrat

CSCI 313

Problem 2

For this question we need to write in a prefix expression for the previous question in the main. Prefix can be seen as faster but it is quite similar to postfix functions. We know the operators are written before their operands. Prefix and postfix are also known to faster modes of operation than infix. So what we need to do is follow the algorithm to evaluate the prefix expression in this question. Having an abstract data type, the operations used in stack are push, which adds an element to the collection. Pop: removes the most recently added element that was not yet removed.

We need to put a pointer X at the end of the operator. Then if character is at X then we need to use the push command to move it to the stack. But if the character at X is an operator of pop, then you need to move the elements from the stack. It can be one to two elements that can be popped. In this question what we are going to deal with is a similar algorithm that was given for postfix. Using a double evaluatePrefix, with a string expression. Then the cases will be start with and then proceed with +, -, *, /. All of these operations will have a declared as stack.push. By the end of all the declarations, the stack will return to the top until the expression has been completed with the correct output.