|  |
| --- |
| Infix to Postfix |
| -expression : string  -postfix : string |
| +InfixtoPostfix (expression : string) : String  +HasGreaterImportance( operator1 : char, operator2 : char) : Int  +IsOperator(C : char) : Bool  +IsOperand(C : char) : Bool  +GetOperatorSignificance ( op : char) : int  +HasGreaterImportance(op1 char, op2 : char) : int |

Implementation

I was able to fully implement a program that converts an infix expression to a postfix expression, this included functions that would decide if a character was a valid letter or number, give an operator significance and order these characters depending on their significance.

Instructions

When the program loads you will be promoted to input an infix expression, an infix expression must obey the following rules: Operations performed right to left, Multiplication and division performed ahead of addition and subtraction, Expressions inside brackets are executed first. An example of a infix expression is (A+B)/(C\*D). To get your Postfix expression you must hit ‘Enter’ after your infix expression, this will then display your original input again, then the postfix expression below this.

To exit the program please press enter after the postfix expression is displayed, or by clicking the ‘X’ at the top of the console.

|  |  |
| --- | --- |
| Infix Expression | Postfix Expression |
| A - B | AB- |
| 3 \* (5+6) | 356+\* |
| a \* (b+c)/(d-e) | abc+\*de-/ |
| (4-6)\*(8+2)/(a-b)(c+d) | 46-82+\*ab-cd+/ |
| A + ((B +C) \* (E – F) – G) / (H – I) | ABC+EF-\*G-+HI-/ |