**Tutorial 10**

**Stacks**

1. Transform the following infix expressions to postfix and prefix:

* 1. (A \* B) / C   
     ***Postfix: A B \* C /   
     Prefix: / \* A B***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***B*** |  | ***C*** |  |
| ***A*** | ***A*** | ***A \* B*** | ***A \* B*** | ***AB / C*** |

* 1. A – (B \* C) + D / E   
     ***Postfix: ((A – (B \* C)) + (D / E)) => A B C \* - D E / +  
     Prefix:***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ***C*** |  |  |  |  |  |  |
|  | ***B*** | ***B*** | ***B \* C*** |  |  |  |  |  |
| ***A*** | ***A*** | ***A*** | ***A*** | ***A - (B\*C)*** |  |  |  |  |

* 1. (X – 5) + (7 \* Z) / V   
     ***Postfix: ( (X - 5) + ( (7 \* Z) / V) ) => X 5 – 7 Z \* V / +   
     Prefix:***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 1. V \* W \* 8 + Y - Z   
     ***Postfix: ( ( ( (V \* W) \* 8 ) + Y ) – Z) => V W \* 8 \* Y + Z -   
     Prefix:***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 1. A / B \* C – D + E   
     ***Postfix: ( ( (A / B) \* C) – D) + E) => A B / C \* D – E +  
     Prefix:***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. Trace and evaluate the following postfix expressions:

* 1. 3 8 + 10 3 - \*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 3 | 8 | + | 10 | 3 | - | \* |  |  |  |  |
| Stack |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 |  |  |  |  |  |  |
|  | 8 |  | 10 | 10 | 7 |  |  |  |  |  |
| 3 | 3 | 11 | 11 | 11 | 11 | 77 |  |  |  |  |

* 1. 2 3 ^ 4 ^

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Expression: | 2 | 3 | ^ | 4 | ^ |
| Stack |  |  |  |  |  |
|  |  |  |  |  |
|  | 3 |  | 4 |  |
| 2 | 2 | 8 | 6 | 4096 |

* 1. 8 2 + 10 - 10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Expression: | 8 | 2 | + | 10 | - | 10 |
| Stack |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* 1. 15 3 / 5 10 – / 100 ^

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 15 | 3 | / | 5 | 10 | - | / | 100 | ^ |
| Stack |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

* 1. 25 4 \* 20 15 - / 2 / 18 +

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 25 | 4 | \* | 20 | 15 | - | / | 2 | / | 18 | + |
| Stack |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

IT2153/IT2352/IT2553/IT2653/IT2852

Data

Structures & Algorithms



1. Write a Python function – transfer(S, T) with no more than 3 lines of code, that transfers all elements from stack S to stack T, so that the element starts at the top of S is the first to be inserted onto T, and the element at the bottom of S ends up at the top or T.

Text

Description automatically generated with low confidence

1. Write a recursive Python function – recEmptyStack(S) for removing all the elements from a stack S.

Graphical user interface

Description automatically generated with low confidence

***-- End of Tutorial --***

AY2020/21 S1 Page 1