**Tutorial 10**

**Stacks**

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

* 1. (A \* B) / C

**Postfix**

Step 1: ((A \* B) / C)

Step 2: ((A B \*) / C)

Step 3: A B \* C /

**Prefix**

Step 1: ((A \* B) / C)

Step 2: ( / (\* A B) C)

Step 3: / \* A B C

* 1. A – (B \* C) + D / E

**Postfix**

( (A – (B \* C) ) + (D / E) )

( (A (B C \*) -) (D E /) +)

A B C \* - D E / +

**Prefix**

( (A – (B \* C) ) + (D / E) )

(+ (- A (\* B C) ) (/ D E) )

+ - A \* B C / D E

* 1. (X – 5) + (7 \* Z) / V

**Postfix**

( (X – 5) + ( (7 \* Z) / V) )

( (X 5 -) ( ( 7 Z \*) V /) +)

X 5 – 7 Z \* V / +

**Prefix**

( (X – 5) + ( (7 \* Z) / V) )

(+ (- X 5) (/ (\* 7 Z) V) )

+ - X 5 / \* 7 Z V

* 1. V \* W \* 8 + Y – Z

**Postfix**

( ( ( (V \* W) \* 8) + Y) – Z)

( ( ( (V W \*) 8 \*) Y +) Z -)

V W \* 8 \* Y + Z –

**Prefix**

( ( ( (V \* W) \* 8) + Y) – Z)

(- (+ (\* (\* V M) 8) Y) Z)

- + \* \* V M 8 Y Z

* 1. A / B \* C – D + E

**Postfix**

( ( ( (A / B) \* C) – D) + E)

( ( ( (A B /) C \*) D - ) E +)

A B / C \* D – E +

**Prefix**

( ( ( (A / B) \* C) – D + E)

(+ (- (\* (/ A B) C) D) E)

+ - \* / A B C D E

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 | 8 | 4096 |  |  |  |  |  |  |

* 1. 8 2 + 10 – 10

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 8 | 2 | + | 10 | - | 10 |  |  |  |  |  |
| Stack |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | 10 |  | 10 |  |  |  |  |  |
| 8 | 8 | 10 | 10 | 0 | 0 |  |  |  |  |  |

Invalid Postfix Expression! (Too many values left on the stack).

* 1. 15 3 / 5 10 – / 100 ^

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 15 | 3 | / | 5 | 10 | - | / | 100 | ^ |  |  |
| Stack |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 10 |  |  |  |  |  |  |
|  | 3 |  | 5 | 5 | -5 |  | 100 |  |  |  |
| 15 | 15 | 5 | 5 | 5 | 5 | -1 | -1 | 1 |  |  |

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Expression: | 25 | 4 | \* | 20 | 15 | - | / | 2 | / | 18 | + |
| Stack |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 15 |  |  |  |  |  |  |
|  | 4 |  | 20 | 20 | 5 |  | 2 |  | 18 |  |
| 25 | 25 | 100 | 100 | 100 | 100 | 20 | 20 | 10 | 10 | 28 |

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.

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

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

AY2020/21 S1 Page 1