Testing:

Assumptions in code:

- Input string is a valid postfix expression - can be converted to infix
- Input string has length >0
- Input string contains only integers and +-* operators

Error can occur when an assumption fails. So testing can be done to validate all assumptions and error handeling when assumption is violated

1. simple string with all 3 operations:

This test is to make sure all the 3 operations are working correctly without overflow issues

1 2 _ 3+4+ 8 6 _ +2 5 _-3 1 _ -8 5 _ +1-7 3 _+ =

1 _ 2 + 3 + 4 + 8 _ 6 - 2 _ 5 - (3 _ 1) + 8 _ 5 - 1 + 7 _ 3 = 104

```
Enter the Postfix Expression: 12*3+4+86*+25*-31*-85*+1-73*+
Length of exp is: 29
Multiplying: 2 & 1
= 2
Adding: 3 & 2
= 5
Adding: 4 & 5
= 9
Multiplying: 6 & 8
= 48
Adding: 48 & 9
= 57
Multiplying: 5 & 2
= 10
Subtracting: 10 from 57
= 47
Multiplying: 1 & 3
= 3
Subtracting: 3 from 47
= 44
Multiplying: 5 & 8
= 40
Adding: 40 & 44
= 84
Subtracting: 1 from 84
= 83
Multiplying: 3 & 7
= 21
Adding: 21 & 83
= 104
The Expression evaluates to: 104
```

2. invalid inputs

This test is to check any characters apart from integers and 3 operators in the input string

45+a- = 4+5 - a

```
Enter the Postfix Expression: 45+a-
Length of exp is: 5
Adding: 5 & 4
= 9
Invalid Input expression. Enter expressions with [0-9+-*x]* only!!!
```

3. Empty expression

This is to handle corner case with no input at all

```
Enter the Postfix Expression:
Length of exp is: 0
Empty expression
```

4. Case of faulty Input string where the stack becomes empty in the middle of parsing

This is to test error handeling if string is not a valid postfix expression as it demands to pop from an empty stack

8+9 = invalid postfix

```
Enter the Postfix Expression: 8+9
Length of exp is: 3
Adding: Invalid Input Expression. Empty Stack State encountered
```

5. input string where the stack's size is not 1 at the end of parsing:

This is to test if at the end of parsing stack size is not equal to 1

45+4 = invalid postfix

```
Enter the Postfix Expression: 45+4
Length of exp is: 4
Adding: 5 & 4
= 9
Invalid Expression. Stack has more than 1 element after evaluation
```