Problem 1

1.

3.

First Method:

Taking the step size of 8 to resize the stack from 8 to 32

K=0 8 -------0

K=1 16-------8

K= 2 24---------16

K=3 32-------24

total copy operations: 8 + 16+24 = 48

Applying the Theory

The Total Copy operations C = N*(N-n)/(2*n)

Taking N = 32, n =8

C = 32*(32-8)/(2*8) = 48

Second Method:

N= 32, n =8

8------0

16 ---8

32 -------16

total copy operations:

<u>for N =32:</u> 8+16 =24

Applying the Theory N-n

For N = 32 , n = 8

Total copy operations K = 32-8 = 24

Problem 3

Test Cases:

Junit Tests are provided in the source code in the class **ArithmeticTermTest**.

1: `( 5.1 * ( ( ( 9 + 8.88 ) ^ ( ( sqrt 4 ) / 6 ) ) - 7 ) )`

Result: `5.1 9 8.88 + 4 sqrt 6 / ^ 7 - *`

`2. ( 4 + ( ( 3 / 5 ) * ( 6 ^ 8 ) ) )`

Result: `4 3 5 / 6 8 ^ * +`

`3. ( 9.2 - ( ( 5 * 3.4 ) ^ ( 6 + 7 ) ) )`

Result: `9.2 5 3.4 * 6 7 + ^ -`

`4. ( ( sqrt 2 ) * ( ( 3 + 8.7 ) / ( 13 + 48.3 ) ) )`

Result: `2 sqrt 3 8.7 + 13 48.3 + / *`

5. `( 4.6 * ( ( 508 + 22 ) / ( ( 45 ) ^ 2 ) ) - 14 )"`

Result: `4.6 508 22 + 45 / 2 ^ * 14 -`

For part 2 of problem 3 , the code from my teammate was included (double evaluate()) and used in the main method and tested with Junit Test in the method `void mergedTest()` in the class ArithmeticTermTest.

Resources:

[1] https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html

[2] Sedgwick Slideset 2: Stack Implementations