**Introduction to Computer Science, IDC Herzliya, 2016**

# HW 3: String and Array Processing

---

Before you start implementing any program, we recommend writing pseudo-code *on paper*, tracing its execution *on paper*, and convincing yourself that you got the logic right.

1. **Parser**

   In Lecture 3.2 we described a parser program. Complete the given `Parser.java` program, and test it using the given `TestParser.java` program.

2. **Expand**

   Write a program, `Expand.java`, that performs the expand operation described in lecture 3.2. Here is an example of the program's execution:

   ```
   % java Expand G2T11C4A5T1G8C2
   GGTTTTTTTTTTTCCCCAAAAATGGGGGGGGCC
   ```

   Your `Expand` implementation should make use of the `Parser`'s services. Assume that the input consists of a valid sequence of (*char*, *n*) pairs, where *n* is a sequence of one or more digits.

3. **Eval**

   Write a program, `Eval.java`, that performs the arithmetic evaluation operation described in lecture 3.2. Here is an example of the program's execution:

   ```
   % java Eval 3+15-2+50
   3+15-2+50 = 66
   ```

   Your `Eval` implementation should make use of the `Parser`'s services. Assume that the input consists of a valid sequence of (*operand*, *operator*) pairs, where *operand* is a sequence of one or more digits, and *operator* is either `+` or `-`.

4. **Statistics**

   Write a program, `Stats.java`, that accepts an array of integer values and computes their average and absolute deviation. The average is the sum of the values divided by the number of values. The absolute deviation is the sum of the absolute deviations of each value from the average, divided by the number of values. For example, if the average is 5, then the absolute deviation of 3 is 2, as is the absolute deviation of 7. Here is an example of the program's execution:

   ```
   % java Stats 1 2 4 6 7
   The average is 4.0. The absolute deviation is 2.0.
   ```

5. **First Repeating Element**

   Write a program, `FindFirstRepeat.java`, that finds and prints the first repeating element in an array of integers. Here is an example of the program's execution:

   ```
   % java FindFirstRepeat 10 5 3 4 3 5 6
   First repeating element is 5

   % java FindFirstRepeat 8 1 3 5 4
   No repeating elements
   ```

   Note: you are not allowed to sort the array.

6. **Frequency Counter**

   We wish to write a program that computes how many times each letter in the alphabet appears in a given text. Assume that the alphabet is the lower-case letters `'a'`,`'b'`,`'c'`,...,`'z'`, and that the text is represented by a command-line string.

   Write a program, `FreqCalc.java`, that consists of a `main` method and a function. The function accepts a string and returns an array of 26 integers which represent letter frequencies, as follows. The first value in the array is the number

of times that the letter `'a'` appears in the string; the second array value is the number of times that `'b'` appears in the string, and so on. The main method gets the input text, calls the function, and prints a histogram, as shown below.

Notes: (1) A string that contains spaces must be enclosed in quotation marks (as shown below); otherwise the operating system will consider it as many separate strings and not as a single string. (2) The input may contain characters that are not in the alphabet. These characters are ignored, and not accounted for.

```
% java FreqCalc "Behold, Arjuna, a million divine forms, with an infinite variety of color and shape.
Behold the gods of the natural world, and many more wonders never revealed before.
Behold the entire cosmos turning within my body, and the other things you desire to see.
I am time, the destroyer of all; I have come to consume the world."

a:***************
b:*****
c:****
d:***************
e:**********************************
f:******
g:***
h:***************
i:******************
j:*
k:
l:************
m:**********
n:*******************
o:*************************
p:*
q:
r:******************
s:***********
t:********************
u:*****
v:*****
w:*****
x:
y:******
z:
```

**Submission:** Before submitting any program, take some time to inspect your code, and make sure that it is written according to our **Java Coding Style Guidelines**. Also, make sure that each program starts with the program header described in the **Homework Submission Guidelines**. Any deviations from these guidelines will result in points penalty.

Submit the following files only:

- `Parser.java`
- `Expand.java`
- `Eval.java`
- `Stats.java`
- `FindFirstRepeat.java`
- `FreqCalc.java`

**Deadline:** Submit your assignment no later than Sunday, December 4, 23:55.