COS126 Prep for Sequence Alignment Assignment

- Group Activity: Calling non-static methods, Edit distance, Recovering the alignment, static constants.

1. Calling non-static methods. Non-static methods will be associated with a data type. Call them using a variable name declared as that data type.

   What does the following code fragment print? (p. 339)

   ```
   String a = "now is ";
   StdOut.println(a.length());
   StdOut.println(a.charAt(4));
   ```

2. Manually computing the edit distance from 2 aligned strings.

   ```
   A A C A G T T A C C                A A C A G T T A C C
   T A A G G T C A - -                T A - A G G T - C A
   1 0 1                              1 0 2
   edit distance:                     edit distance:
   ```

3. Manually computing the edit distance from 2 strings.

   Suppose you have two matching characters. x = ”A”, y = ”A” The bottom row and rightmost column are your ”base cases.” The bottom row represents no more x characters. The rightmost column represents no more y characters.

   ```
   x\y A -
   A
   -
   ```

   Suppose you have a mismatch in the last two characters. x = ”A”, y = ”C”.

   ```
   x\y C -
   A
   -
   ```

Fill in the matrix for the strings: x = ”ATAG”, y = ”TAC”

```
x\y T A C -
A         8
T         6
A         4
G   5 3 1 2
-   6 4 2 0
```

4. Recover the alignment for x = ”ATAG”, y = ”TAC”, using the completed matrix from the previous part.

```
x y penalty
```

5. Recover the alignment for x = ”bb”, y = ”bbb” using the following matrix.

```
x\y b b b -                    x y penalty
b   2 0 2 4
b   4 2 0 2
-   6 4 2 0
```

6. static constants are constants which are declared and initialized inside the class block, but outside any methods. These constants are then available to any method in the class, including `main()`.

   To make your code easier to read, and to avoid hardwiring constants, declare the three penalty values as static constants.

```
public class EditDistance {
    private  static  final  int  GAP  =  2;
    private __________________________________________
    private __________________________________________

    // methods . . .
}
```

Recommended Book exercises for analysis: 4.1.10, 4.1.18, 4.1.20, 4.1.27