Arrays

CMPT220L
Due on Mar 25, 2022 by 11:59 PM
Points: 100

Problems

1. *(Count single digits)* Write a program that generates 100 random integers betweeen 0 and 9 and displays the count for each number. (*Hint:* Use an array of 10 integers, say `counts`, to store the counts for the numbers of 0s, 1s, ..., 9s.)

2. *(Area of a polygon)* Write a program that prompts the user to enter the points of a convex polygon and display its area. Assume that the polygon has six end points and the points are entered clockwise. For the definition of a convex polygon, see `http://www.mathopenref.com/polygonconvex.html`. *Hint*: the total area of a polygon is the sum of the areas of the small triangles as shown in Figure 1.

3. *(Occurrences of each digit in a string)* Write a method that counts the occurrences of each digit in a string using the following header:

   ```
   public static int[] count(String s)
   ```

   The method counts how many times a digit appears in the string. The return value is an array of ten elements, each of which holds the count for a digit. For example, after executing `int[] counts = count("12203AB3")`, `counts[0]` is 1, `counts[1]` is 1, `counts[2]` is 2, and `counts[3]` is 2. Write a test program that prompts the user to enter a string and displays the number of occurrences of each digit in the string.

   ```
   Enter a string: 23231adbc1234
   Digit 1 occurs 2 times
   Digit 2 occurs 3 times
   Digit 3 occurs 3 times
   Digit 4 occurs 1 time
   ```

4. *(Algebra: 2×2 matrix inverse)* The inverse of a square matrix A is denoted A^{-1}, such that $A \times A^{-1} = I$, where I is the identity matrix with all 1s on the diagonal and 0 on all other cells. For example, the

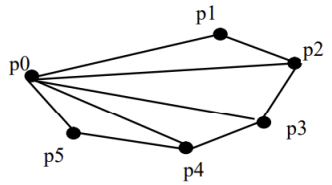

Figure 1: A convex polygon can be divided into small non-overlapping triangles.

inverse of matrix $\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ is $\begin{bmatrix}-0.5 & 1\\ 1.5 & 0\end{bmatrix}$, i.e.,

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \times \begin{bmatrix}-0.5 & 1\\ 1.5 & 0\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \tag{1}$$

The inverse of a 2×2 matrix A can be obtained using the following formula:

$$A = \begin{bmatrix}a & b\\ c & d\end{bmatrix} \qquad A^{-1} = \frac{1}{ad - bc}\begin{bmatrix}d & -b\\ -c & a\end{bmatrix} \tag{2}$$

Implement the following method to obtain an inverse of the matrix:

```
public static double[][] inverse(double[][] A)
```

The method returns `null` if *ad–bc* is 0.

Write a test program that prompts the user to enter *a*, *b*, *c*, *d* for a matrix, and displays its inverse matrix. Here is a sample run:

```
Enter a, b, c, d: 1 2 3 4
-2.0 1.0
1.5 -0.5

Enter a, b, c, d: 0.5 2 1.5 4.5
-6.0 2.6666666666666665
2.0 -0.6666666666666666

Enter a, b, c, d: 1 2 3 6
No inverse matrix
```

Submission

Make sure you create one Java file per project. Place your `.java` files under the corresponding folder in your local copy of the GitHub repository, commit and push it to the remote repository. Make sure that the professor has access to the repository (`jfac65-marist`).

```
cmpt220lastname\
   hw06\
      Problem1.java
      Problem2.java
      Problem3.java
      Problem4.java
```