ACS-1904 Winter 2024
Assignment #1
Due by Sunday, February 4 at 11:59 pm

- Submit your `.java` file(s) via Nexus
- Include your name and student number as a comment

Part1: Loading the data and performing the initial calculations:

Write a Java program called `SpeedCameras` that reads data from a file called `A1Data.txt`. You can find `A1Data.txt` in the content section of Nexus and linked from the class schedule. The file contains a list of speed camera records consisting of the intersection code, the posted speed limit, the last 10 speed readings taken by the remote sensor, and the number of vehicles that have passed through the intersection in each of the last three hours. Store each of the intersection codes, posted speed limits, speed sensor data and traffic volume data in **parallel arrays**, where each array is related. Use an array of *`Strings`* for the intersection codes and an array of type *`int`* for the posted speed limits. Use **2D arrays** for the speed sensor data, and traffic volume data (i.e. a 2D array for the last 10 speed sensor readings, and another for the last 3 traffic volume readings) both of these 2D arrays are of type int.

Add four more arrays for the following statistics:

1) Average speed: drop the highest and lowest speeds then calculate the average. The average should be of type double.
   a) Also, note that the original order of the speeds must be maintained. Even though they are not included in the summary when printed it is still necessary to keep the original order so that the correct speeds will be removed as new speeds are added.
2) Average volume: the average of the traffic volume readings. This average is also of type double.
3) The Speed Score. The intersection's speed score is calculated with this formula;

   `Average speed/speed limit * average traffic volume.`

4) Rank: a ranking from 1 to n with 1 being the worst (highest) score. Note that the intersection records should not be sorted into rank order. (see the sample output for clarification)

The data file is arranged as follows. Note that the labels in the first row are not present in the file they are included here for information only.

ID	Limit	Last 10 speed sensor readings										Last traffic volume readings		
Por-Mai	50	45	46	56	56	50	70	54	54	59	49	483	517	533
StM-Fer	60	70	59	54	52	78	60	77	69	59	73	488	495	454
StA-Fer	60	77	83	61	75	67	54	53	65	76	58	545	519	485
KgE-Sar	70	66	82	82	75	94	89	92	87	86	61	548	450	542
Nes-StJ	50	52	49	58	47	49	61	46	58	57	67	430	440	462
Pem-Bis	60	84	80	74	72	53	80	79	82	64	64	443	414	532
Wav-Ken	80	71	79	68	68	108	80	92	74	88	75	458	449	505

When the program runs, display the data summarized in the following format:

```
Intersection      Limit Avg Speed   V1     V2     V3     Avg Vol  Score       Rank
Por-Mai              50     53.00   483    517    533    511.00   541.66      4
StM-Fer              60     65.13   488    495    454    479.00   519.91      5
StA-Fer              60     66.63   545    519    485    516.33   573.35      3
KgE-Sar              70     82.38   548    450    542    513.33   604.08      1
Nes-StJ              50     53.88   430    440    462    444.00   478.41      6
Pem-Bis              60     74.38   443    414    532    463.00   573.93      2
Wav-Ken              80     78.38   458    449    505    470.67   461.11      7
```

Note that the ***Avg Speed*** column contains the overall average of 8 of the last 10 speed sensor readings (don't forget to drop the highest and lowest speeds) and should be displayed in 2 decimal places. The ***Avg Vol*** and ***Score*** columns should likewise be to two decimal places.

Part 2: Adding data

The user will be given the option to add a stat or quit. This process will continue until the user chooses to quit. The user's choice should not be case sensitive i.e. entering ***A*** or ***a*** should initiate the add a stat process.

- If the choice is to add a stat the system prompts the user for
    - the intersection code of the stat to be added,
        - note if the intersection code is not found in the list the user should be forced to re-enter the code until they enter an intersection that is in the list.
    - the type of stat to be added, either speed or volume,
    - and the new score.
    - The new stat will be added to the end of the list (index n – 1), all of the other stats must be shifted down one index and the stat currently in index 0 will be lost.
- The system re-calculates all of the averages and the rank and displays the summary again.

When quit is chosen the program will continue on to the "*end of program message*".

Include methods `calculateAverageSpeed(…)`, `calculateAverageVolume(…)`, `calculateSpeedScore(…)`, `calculateRank(…)`, `getIndexToEdit(…)`, `addStat(…)`, and `displaySummary(…)`.

You can add other utility methods as needed. For example, I added a method to shift the values in an array (part of the adding a stat process)

Notes:

- Use the size of the sample data for *initializing* array lengths
    - The arrays are all full
- You can use .length or you may use *static final …* constants in your methods to control the number of iterations of for loops
- To format a double to 2 decimal places, use `String.format("%.2f", value)`
- ArrayLists may not be used, but you may use other utilities from the Java Class Libraries.
- Don't use break except in switch statements.
- Your code should take into account the possibility that the data file might be empty, and that

the user may not want to make any changes once the original data has been read, calculations made and the summary displayed.

- Use auxiliary arrays as needed.
- The size of the arrays and tables does not need to be scalable.
- while and do-while loops should have the form while(<logical expression>, where <logical expression> is something like repeat != “yes”. Note that fin.hasNext() is an acceptable logical expression.
- All user input should be non-case sensitive.

Sample output (text in **blue** is user input): NOTE, your code must work for possible inputs.

```
Speed Camera Statistics
*************************************************************************

Intersection      Limit Avg Speed   V1     V2     V3     Avg Vol  Score       Rank
Por-Mai              50     53.00   483    517    533    511.00   541.66      4
StM-Fer              60     65.13   488    495    454    479.00   519.91      5
StA-Fer              60     66.63   545    519    485    516.33   573.35      3
KgE-Sar              70     82.38   548    450    542    513.33   604.08      1
Nes-StJ              50     53.88   430    440    462    444.00   478.41      6
Pem-Bis              60     74.38   443    414    532    463.00   573.93      2
Wav-Ken              80     78.38   458    449    505    470.67   461.11      7

******* Edit Records *******

Choose an option:
A - Add a stat
B - Quit

a
Enter the intersection code for the record to edit.
por-mai

Select:
S: add a speed
V: add a volume
s
Enter the new stat value.
62
Intersection      Limit Avg Speed   V1     V2     V3     Avg Vol  Score       Rank
Por-Mai              50     55.00   483    517    533    511.00   562.10      4
StM-Fer              60     65.13   488    495    454    479.00   519.91      5
StA-Fer              60     66.63   545    519    485    516.33   573.35      3
KgE-Sar              70     82.38   548    450    542    513.33   604.08      1
Nes-StJ              50     53.88   430    440    462    444.00   478.41      6
Pem-Bis              60     74.38   443    414    532    463.00   573.93      2
Wav-Ken              80     78.38   458    449    505    470.67   461.11      7
Choose an option:
A - Add a stat
B - Quit

a
Enter the intersection code for the record to edit.
pem-bis

Select:
S: add a speed
```

```
V: add a volume
s
Enter the new stat value.
55
Intersection      Limit Avg Speed   V1     V2     V3     Avg Vol  Score       Rank
Por-Mai              50     55.00   483    517    533    511.00   562.10      3
StM-Fer              60     65.13   488    495    454    479.00   519.91      5
StA-Fer              60     66.63   545    519    485    516.33   573.35      2
KgE-Sar              70     82.38   548    450    542    513.33   604.08      1
Nes-StJ              50     53.88   430    440    462    444.00   478.41      6
Pem-Bis              60     71.00   443    414    532    463.00   547.88      4
Wav-Ken              80     78.38   458    449    505    470.67   461.11      7
Choose an option:
A - Add a stat
B - Quit

a
Enter the intersection code for the record to edit.
wav-ken

Select:
S: add a speed
V: add a volume
v
Enter the new stat value.
550
Intersection      Limit Avg Speed   V1     V2     V3     Avg Vol  Score       Rank
Por-Mai              50     55.00   483    517    533    511.00   562.10      3
StM-Fer              60     65.13   488    495    454    479.00   519.91      5
StA-Fer              60     66.63   545    519    485    516.33   573.35      2
KgE-Sar              70     82.38   548    450    542    513.33   604.08      1
Nes-StJ              50     53.88   430    440    462    444.00   478.41      7
Pem-Bis              60     71.00   443    414    532    463.00   547.88      4
Wav-Ken              80     78.38   449    505    550    501.33   491.15      6
Choose an option:
A - Add a stat
B - Quit

b

end of program
```

Submit your Java file (`SpeedCameras.java`) via Nexus.