

Programming Assignment #3: Vehicle Track Performance Tests

Purpose: The purpose of this programming assignment is to give the student experience in using functions, arrays, and strings. Arrays will be used to store and process data and the graphing capabilities of Excel will be used to graph results calculated by the C++ program.

Reference: The PowerPoint presentation on data files on the instructor's website includes an example of graphing the output of a C++ program using Excel.

Data file description:

Four data files (CAR1.DAT, CAR2.DAT, CAR3.DAT, and CAR4.DAT) are available on the instructor's web page. Each file contains track results for a test vehicle. The results are in the following format:

- The test vehicle number (line 1).
- The test date (line 2).
- The test driver (line 3).
- Time, t, in seconds and distance, x, in feet (one pair of (t,x) values per line for all other lines)

As an example, the contents of CAR1.DAT are shown below:

```
Test vehicle: Car #1
Test date: October 26, 2007
Driver: D. Myers
0.0  0.0
0.2  5.5
0.4  15.9
0.6  30.9
0.8  49.7
1.0  71.8
1.2  97.1
1.4  125.4
1.6  156.6
1.8  190.4
2.0  226.6
2.2  265.3
2.4  306.1
2.6  348.9
2.8  393.5
3.0  439.9
3.2  488.0
3.4  537.7
3.6  588.8
3.8  641.2
4.0  694.6
4.2  749.2
4.4  804.7
4.6  861.2
4.8  918.4
5.0  976.2
```

Determining velocity and acceleration using distance and time values:

Velocity, v , and acceleration, a , can be determined using values of distance, x , and time, t , as follows. By definition:

$$v = \frac{dx}{dt} \quad \text{and} \quad a = \frac{dv}{dt}$$

These equations can be approximated by:

$$v = \frac{\text{change in distance}}{\text{change in time}} \quad \text{and} \quad a = \frac{\text{change in velocity}}{\text{change in time}}$$

For the distance and time values in the data files provided, let's assume that $v = 0$ and $a = 0$ at the first time value (0.0). We could then calculate velocity (in ft/s) and acceleration (in ft/s/s) at the second time value using:

$$v1 = \frac{x1 - x0}{t1 - t0} \quad \text{and} \quad a1 = \frac{v1 - v0}{t1 - t0}$$

Similarly at the third time value:

$$v2 = \frac{x2 - x1}{t2 - t1} \quad \text{and} \quad a2 = \frac{v2 - v1}{t2 - t1}$$

For the data file CAR1.DAT, this would yield:

$$v1 = \frac{5.5 - 0.0}{0.2 - 0.0} = \frac{5.5 \text{ ft}}{0.2 \text{ s}} = 27.5 \text{ ft / s}$$

$$a1 = \frac{27.5 - 0.0}{0.2 - 0.0} = \frac{27.5 \text{ ft / s}}{0.2 \text{ s}} = 137.5 \text{ ft / s}^2$$

Similarly:

$$v2 = \frac{15.9 - 5.5}{0.4 - 0.2} = \frac{10.4 \text{ ft}}{0.2 \text{ s}} = 52.0 \text{ ft / s}$$

$$a2 = \frac{52.0 - 27.5}{0.4 - 0.2} = \frac{24.5 \text{ ft/s}}{0.2 \text{ s}} = 122.5 \text{ ft/s}^2$$

The remaining values could be determined in a similar manner. It might also be more practical to express v in mph that in ft/s. A simple conversion factor can be used to convert v in ft/s to mph.

A table could now be created which includes t , x , v (in ft/s), v (in mph) and a . This table for CAR1.DAT might look like:

t (s)	x (ft)	v (ft/s)	v (mph)	a (ft/s ²)
0.0	0.0	0.0	0.0	0.0
0.2	5.5	27.5	18.8	137.5
0.4	15.9	52.0	35.5	122.5
0.6	30.9	75.0	51.5	115.0
5.0	976.2	289.0	197.0	15.0

Program Requirements:

Write a C++ program that will accomplish the following:

1. The program should ask the user which data file is to be used (give an example name and path), read the file name as a string, and open the file using a string variable.
2. The program should read the vehicle number, date, and driver as strings. Also read the values of t and x from the file and store them in arrays. Assume that the data files will contain a maximum of 30 (t, x) data points.
3. Define arrays for velocity (in ft/s), velocity (in mph), and acceleration (in ft/s²). Use a separate **function** to calculate each of these arrays. For example, the function to find velocity in f/s might be called using **Velocity (t,x,v,size)** where t and x are input arrays, v is an output array, and size is the number of data points in each array.
4. Write a general function that can be used to find the average of any array. Call this function twice in order to find average velocity (in mph) and average acceleration (in ft/s² or ft/s/s). For example, the function might be called as follows:

Average = ArrayMean(A, Asize); //function call to find mean value in array A

5. Write a general function that can be used to find the maximum of any array. Call this function twice in order to find maximum velocity (in mph) and maximum acceleration (in ft/s² or ft/s/s). For example, the function might be called as follows:

Max = ArrayMax(A, Asize); //function call to find max value in array A

6. The output of the program should be sent to a data file (the name of the data file should be specified by the user, but give an example name and path). The output should include:
 - The test vehicle number
 - The test date
 - The driver
 - The average velocity in mph
 - The average acceleration in ft/s/s
 - The maximum velocity in mph
 - The maximum acceleration in ft/s/s
 - Values for time, distance, velocity (ft/s), velocity (mph), and acceleration (ft/s/s) for each of the data points in the input data files (i.e., 5 numbers per line).

The file should **include commas** between each numeric value so that the file can be imported into Excel as a **“Commas Delimited File”**. The output data file for CAR1.DAT might look as follows:

As an example, the output file generated for the input CAR1.DAT is shown below:

```

Test vehicle: Car #1
Test date: October 26, 2007
Driver: D. Myers
Average velocity = aaa mph
Average acceleration = bbb ft/s/s
Maximum velocity = ccc mph
Maximum acceleration = ddd ft/s/s
0.0,0.0,0.0,0.0,0.0
0.2,5.5,27.5,18.8,137.5
0.4,15.9,52.0,35.5,122.5
0.6,30.9,75.0,51.5,115.0
.
.
.
5.0,976.2,289.0,197.0,15.0

```

7. Run your program for any two of the data files.
 - Import each of the output data files into Excel.
 - Add titles to each column, lines, numeric formatting, etc., to enhance the appearance and readability of the data.
 - Form four graphs: x vs t , v vs t (in ft/s), v vs t (in mph), and a vs t .
 - In addition to standard report requirements, be sure to include a printout of your program, a printout of all 4 input data files, a printout of all 4 output data file, and the spreadsheet table and graphs for the two files that you selected.

Extra Credit Suggestions: (for a maximum of 10 additional points on the program grade)

1. Use additional functions. For example, a function could be used to create an output data file.
2. Give the user the option of specifying standard or SI units (user selects A or B below):
 - A) distances in the data file are in feet: then calculate v (in ft/s), v (in mph), and a (in ft/s/s)
 - B) distances in the data file are in meters: then calculate v (in m/s), v (in km/h), and a (in m/s/s)
3. Instead of using five one-dimensional arrays for t , x , v (in ft/s), v (in mph), and a (in ft/s/s), use a single two-dimensional array (5 by 30).
4. Expand the idea above to use a three-dimensional array where the third dimension corresponds to the input data file number (1, 2, 3, or 4). Read all four data files when the program starts to fill the array. The program output should then be automatically sent to the four output data files. No user input would be required in running the program.
5. Use your imagination!