C++ Coding Exercise

This problem will require you to write a C++ program that reads an input file ('input.csv') and writes out a new file calculated from the inputs. We will evaluate your submission based on the following criteria:

- Correctness. The program should run properly on Linux. We have provided the skeleton for coding.
- Design and readability
- Tests. Two cases are provided. We welcome more cases (no larger than 2MB in total) from you.
- Use of modern C++14/17/20 Idioms
- Use of C++ templates where appropriate

Part 1:

Please aim to spend about 1-2 hours to complete this exercise.

Input:

The input file represents a very simplified stream of trades on an exchange.
Each row represents a trade. The data can be thought of as a time series of values in columns:

<TimeStamp>,<Symbol>,<Quantity>,<Price>

Although the provided input file is small, the solution should be able to handle a source dataset well beyond the amount of memory and hard disk space on your machine.

Definitions:

- TimeStamp is value indicating the microseconds since midnight.
- Symbol is the 3 character unique identifier for a financial instrument.
- Quantity is the amount traded
- Price is the price of the trade for that financial instrument.

Safe Assumptions:
- TimeStamp is always for the same day and won't roll over midnight.
- TimeStamp is increasing or same as previous tick (time gap will never be < 0).
- Price - our currency is an integer based currency. No decimal points. - Price - Price is always > 0.

Example: here is a row for a trade of 10 shares of aaa stock at a price of 12
1234567,aaa,10,12

Problem:

Find the following on a per symbol basis:

1. Maximum time gap time gap = Amount of time that passes between consecutive trades of a symbol, if only 1 trade is in the file then the gap is 0.
2. Total Volume traded (Sum of the quantity for all trades in a symbol).

3. Weighted Average Price. Average price per unit traded not per trade. Result should be truncated to whole numbers.

Example: the following trades

20 shares of aaa @ 18
5 shares of aaa @ 7
Weighted Average Price = ((20 * 18) + (5 * 7)) / (20 + 5) = 15

4. Max Trade Price.

Your design should allow extension to additional numerical variables computed for each symbol.

Commands:

$ ./Exercise input.csv output.csv

Output:

Your solution should produce a file called 'output.csv'. This file should be a comma separate file with this format:
<symbol>,<MaxTimeGap>,<Volume>,<WeightedAveragePrice>,<MaxPrice>

The output should be sorted by symbol ascending ('aaa' should be first).

Sample Input:

```
52924702,aaa,13,1136
52924702,aac,20,477
52925641,aab,31,907
52927350,aab,29,724
52927783,aac,21,638
52930489,aaa,18,1222
52931654,aaa,9,1077
52933453,aab,9,756
```

Sample Output:

```
aaa,5787,40,1161,1222 aab,6103,69,810,907
aac,3081,41,559,638
```

Please send your source code and output.csv in a zipped git repo back for evaluation when complete. Include the amount of time you spent working on the solution.