Objective

Time Limit: 24 hours. We expect this assignment to take 1-3 hours of dedicated effort.

Your program will analyze trades that a hypothetical trading firm makes in some stocks and output paired trades.

The trades file is a comma-separated values (.csv) file containing a header and list of trades. A trade contains the SIDE (whether the firm B(ought) or S(old) the SYMBOL), the PRICE of the transaction, and QUANTITY of shares executed. TIME in this file is chronological. All prices will have at most two decimal places.

```
TIME,SYMBOL,SIDE,PRICE,QUANTITY

2,ABC,B,10.06,500

4,DEF,S,35.99,200

4,ABC,S,10.07,200

5,ABC,S,10.07,300
```

Download sample trades file

Task Description

Form opening-closing trade pairs in a first-in-first-out (FIFO) manner:

- Every symbol begins with zero inventory. The first execution will always be an opening trade as it opens new inventory. In the example trades file above, we see an opening trade for symbol "ABC" that creates net inventory of 500 at time 2.
- If trades occur which increase the magnitude of net inventory (e.g., if another buy were to occur for ABC), these new trades are also opening trades and should be maintained in a FIFO structure. The firm can sell short (i.e. take a negative position) in any symbol.
- Any trade that reduces the magnitude of net inventory (e.g., the 200 share sell at time = 4) will be a closing trade and will pair off with the first available opening trade. The quantity of inventory paired off will be the minimum of the shares executed by the paired opening and closing trades. In this example, this paired quantity is 200 shares since 200 < 500.

If the closing trade is smaller than the paired opening trade, then the opening trade’s open inventory should be reduced but the trade should maintain its FIFO position. If the closing trade is equal in size to the paired opening trade, then the opening trade is completely consumed. If the closing trade is bigger than the paired opening trade, then it consumes the entire opening trade and proceeds to pair against the next opening trades. Finally, if the closing trade is bigger than the entire open inventory, whatever is left over from the closing trade actually “flips” the inventory and creates a new opening trade on the opposite side.

Output

For each paired trade, compute the profit and loss (PNL), which is the product of the paired quantity and the per-share PNL (difference between sell price and buy price). Write all paired trades in chronological order by close time to standard output. There may be some nonzero inventory at the end. Only print paired trades.

After all paired trades have been printed, print the cumulative realized PNL of all paired trades to standard output. This value should be the sum of all individual PNLs.

The output of your program should resemble:

```
OPEN_TIME,CLOSE_TIME,SYMBOL,QUANTITY,PNL,OPEN_SIDE,CLOSE_SIDE,OPEN_PRICE,CLOSE_PRICE

2,4,ABC,200,2.00,B,S,10.06,10.07

2,5,ABC,300,3.00,B,S,10.06,10.07

5.00
```

Submission Instructions

Please implement your solution in C++17 or Python 2.7 and submit your code in a single source file (.cpp or .py). Your program will take a single command line argument that specifies the path of the trades file we will use to grade your submission. You are allowed a single submission. Your code will be evaluated based on **correctness**,**efficiency** and **speed** of the program. Passing solutions will be reviewed by human graders for **thoughtfulness**and the use of **good programming practices**.

C++

We will compile your code using GCC 5.4.0 and run your binary on 64-bit Ubuntu 16.04 LTS. If you choose to develop on an environment different from ours, you are responsible for writing your solution in a portable fashion.

How we will invoke your program:

```
g++ -std=c++17 -O3 -o vatic_code_test *.cpp && ./vatic_code_test [trade file]
```

Python

We will run your code using Python 2.7.12 on 64-bit Ubuntu 16.04 LTS. Modules in the Python Standard Library will be available for your use. **No other third-party packages (e.g. numpy, pandas, etc.) are allowed.**

How we will invoke your program:

```
python -O your_program.py [trade file]
```