DISCLAIMER: We are using actual stock market data for academic purposes only. I am NOT in any way suggesting that you invest or trade in the stock market and other financial markets. Stock market investing is RISKY, and you may lose money!

CCPROG2 MP SPECIFICATIONS PART 3 [25 points]
AY2024 Term 2

サルバドール・フロランテ

- Part 3 covers the application of **`struct`** data type in combination with previous ideas learned.
- Please be reminded again that you are NOT allowed to use functions which were not discussed in class (unless specified otherwise).
- The data source for Challenges 7 and 8 is one of the output text files produced in Challenge #2. The file **`AC.txt`** is used as an example input file for the expected output mentioned in the remainder of this document.
- Question? Press this link and post your question in the Canvas discussion thread: Q&A on the Machine Problem (MP).

IMPORTANT NOTE: use double data type, not float, for real numbers

Philippine Stock Market Data: (some simple) Analysis

Challenge #6: Create a struct Data Type for Representing and Storing SHD [5 pts].

As you have already learned from the previous MP Challenges, information about a particular stock (like AC) stored in a stock historical data (SHD) file is comprised of:

- Stock Name/Code (string, for example "AC")
- Number of rows of historical data transactions (integer, for example, for AC it is 1216 as encoded in the provided AC.txt file; note that this number varies depending on the stock, i.e., it is not always 1216!)
- Rows of daily historical data, where each row encodes the following:
    - date (string in "MM/DD/YYYY" format)
    - open, high, low, close prices (double, referred to as OHLC)
    - volume (double)

Your task for this challenge is to DECLARE a **`struct`** data type using combinations of what you learned in Chapters 1 (Arrays), Chapter 2 (Strings) and Chapter 3 (struct) for representing and storing SHD for just **one** company.

Encode your declarations in the accompanying skeleton header file **`GROUPNUMBER-C6.h`**. Comply with the instructions stated in the header file. Make sure that you accomplish this challenge properly because your header file is crucial in solving the succeeding challenges!

Hint: you can actually compile just the header file to determine if there are no warning or syntax errors. Try it by

```
C:\CCPROG2> gcc -Wall 01-C6.h
```

Replace 01 with your own group number.

DELIVERABLE: Submit only one file, i.e., your **`GROUPNUMBER-C6.h`** header file via Canvas before the indicated submission deadline.

Challenge #7: Compute Profit/Loss [10 pts].

"Buy low, sell high." is a famous phrase known to investors and traders. It means that to make *profit* in the stock market, a share of stock should be bought at a low price, and then sold later at price higher than the buy price. A stock that is sold at a price lower than the buy price will result into a *loss*.

Example Trading Scenarios:
Trade Scenario #1: An investor bought a stock for ₱20.00 per share. After about six months, the stock was sold at ₱22.50 per share. This results to a realized profit of ₱2.50 per share computed as 22.50 – 20.00 = 2.50. We also say that the investor gained a *profit* of 12.5% which was computed as (22.50 – 20.00)/20.00 * 100 = 12.50%.

Trade Scenario #2: A newbie trader bought a stock for ₱20.00 per share. On the next day, the stock price plummeted (declined) due to bad news about the company. The trader panicked and sold the stock at ₱18.25 per share. In this case, the trader had a realized *loss* of ₱-1.75 per share (take note of the negative sign) which was computed as 18.25 – 20.00 = -1.75. We also say that the trader incurred a loss of -8.75% which was computed as (18.25 – 20.00)/20.00 * 100 = -8.75%.

In the investing world, the symbol **P/L** is used to mean Profit Loss, and **%P/L** means Percent Profit Loss. In Scenario #1, the P/L is ₱2.50, and the corresponding %P/L is 12.50%. For Scenario #2, the P/L is ₱-1.75 and the %P/L is -8.75%.

Your task for this challenge is to write a program that will compute the P/L and %P/L values given the buy date and sell date as inputs. Take note that your program should only compute the P/L and %P/L values when stock data for these two dates are found. Use the closing price as the price per share of stock.

Read and follow the instructions in the accompanying skeleton file **GROUPNUMBER-C7.c** to produce a C program that will:

1. Input via **scanf()** the contents of the input file and store them into the structure variable for a company's stock historical data. Use input redirection like in the previous challenges.
2. Search the stock historical data corresponding to the buy and sell dates. **You must implement and use a Binary Search algorithm with buy date and sell date as search keys for this purpose.** WARNING: Your solution will automatically be considered incorrect if you used Linear Search with a corresponding score of 0/10 for this challenge!
3. Compute the P/L and %P/L values.
4. Generate an output that will show the search results, and the P/L and %P/L values based on the buy date and sell date. The output should comply with the following format

```
<stock symbol>
<number of rows of stock historical data>
**TEST-CASE-<number>**
BUY DATE <buy date> FOUND IN INDEX <index>! BUY PRICE = <close price>
SELL DATE <sell date> FOUND IN INDEX <index>! SELL PRICE = <close price>
P/L = <value>
%P/L = <value>
```

Refer to the accompanying files **C7-AC-EXPECTED.txt** which contain the expected output using **AC.txt** as input file. Refer also to the buy dates and sell dates encoded in the **main()**. function. The expected PARTIAL output using **AC.txt** data is also shown in the following textbox for your immediate reference.

```
AC
1216

**TEST-CASE-1**
BUY DATE 01/05/2015 FOUND IN INDEX 1215!  BUY PRICE = 675.17
SELL DATE 12/27/2019 FOUND IN INDEX 0!  SELL PRICE = 760.89
P/L = 85.72
%P/L = 12.70

**TEST-CASE-2**
BUY DATE 01/04/2016 FOUND IN INDEX 975!  BUY PRICE = 732.32
SELL DATE 01/04/2017 FOUND IN INDEX 727!  SELL PRICE = 736.19
P/L = 3.87
%P/L = 0.53
   :      :          :         :         :         :          :
   :      :          :         :         :         :          :
**TEST-CASE-5**
BUY DATE 12/10/2019 FOUND IN INDEX 11!  BUY PRICE = 789.47
SELL DATE 12/11/2019 FOUND IN INDEX 10!  SELL PRICE = 765.25
P/L = -24.22
%P/L = -3.07
```

Make sure that there are NO extraneous values in the output of your program. Your program should produce the same result as the expected output files using the same input SHD text files. Non-compliance in the output format will be considered as a logical error which will result in point deductions.

DELIVERABLES: Submit FOUR files via Canvas before the indicated submission deadline

1. C header file named as `GROUPNUMBER-C6.h`.
2. C source file named as `GROUPNUMBER-C7.c`.
3. Redirected output text file named as `GROUPNUMBER-C7-AC-OUTPUT.txt` using `AC.txt` as input file
4. Redirected output text file of the bbtest comparison result named as `GROUPNUMBER-C7-AC-BBTEST.txt`.

Challenge #8: Buy/Sell Signal [10 pts].

Traders rely on technical "signals" based on stock historical data (date, OHLC and Volume) for deciding when to buy or when to sell a stock. For this challenge, you'll implement a function that will output buy/sell signals using a strategy referred to as "*moving average (MA) crossovers".* It relies on two moving averages, one a short-term MA (for example: 50-day SMA) and the second a long-term MA (for example 200-day SMA). A **buy signal** is generated when the short-term MA crosses **above** the longer-term MA. This is called "golden cross". On the other hand, a **sell signal** is generated when the short-term MA crosses **below** the longer-term MA. This is in turn called "death cross".

Watch and learn from the following video (from time 2:22 to 3:30) for a visual explanation of the MA crossover strategy.
https://youtu.be/5rMkQurfxrE?t=142

Read and follow the instructions in the accompanying skeleton file `GROUPNUMBER-C8.c` to produce a C program that will:

1. Input via `scanf()` the contents of the input file and store them into the structure variable for a company's stock historical data. Use input redirection like in the previous challenges.
2. Compute the short-term and long-term simple moving averages, and the signal for the day, i.e., is it a 'B' (buy) or a 'S' (sell)? Note that it is a buy if the short-term moving average is **higher** than the long-term moving average, otherwise, it is a sell.
3. Generate an output with the following format:

```
<stock symbol>
<number of rows of stock historical data>
**TEST-CASE-<number>**
mst = <number>-days, mlt = <number>-days
count = <number>
<counter> <date> <short term MA> <long term MA> <BUY or SELL>
    :        :         :              :              :
<counter> <date> <short term MA> <long term MA> <BUY or SELL>
```

Refer to the accompanying files `C8-AC-EXPECTED.txt` which contain the expected output using `AC.txt` as input file. The short term and long term days are encoded in the **main()** function.

The expected PARTIAL output using `AC.txt` data is also shown in the following textbox for your immediate reference.

```
AC
1216

**TEST-CASE-1**
mst = 9-days, mlt = 20-days
count = 1197
  1   02/03/2015     694.11     688.95    BUY
  2   02/04/2015     697.93     690.54    BUY
  3   02/05/2015     698.36     691.61    BUY
  4   02/06/2015     699.44     692.99    BUY
  :       :              :          :       :
  :       :              :          :       :
```

Make sure that there are NO extraneous values in the output of your program. Your program should produce the same result as the expected output files using the same input SHD text files. Non-compliance in the output format will be considered as a logical error which will result to point deductions.

DELIVERABLES: Submit FOUR files via Canvas before the indicated submission deadline

1. C header file named as **`GROUPNUMBER-C6.h`**.
2. C source file named **`GROUPNUMBER-C8.c`**.

Last set of challenges in Part 4…