

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 1 [20 points]
AY2024 Term 2

サルバドル・フロランテ

GENERAL INSTRUCTIONS

1. Honesty Policy and Honor Code apply. **Cheating is a major offense and will result in a final grade of 0.0.**
2. Carefully read, understand and comply with the requirements and instructions stated in the MP Specifications document and in the accompanying skeleton codes.
3. You will be presented with several challenge problems. Solve the challenges by formulating your algorithms, and by encoding your solutions using the given skeleton codes.
4. You are NOT allowed to use pre-defined library functions that we did not discuss in class (unless specified otherwise)
5. Make sure that there are no warnings, no syntax errors and no logical errors in your solutions. Subject your solution to exhaustive testing to ensure that it is logically correct. The following scoring and deduction system will be applied.
 - A perfect score will be awarded for a compliant and correct solution.
 - A solution with logical error/s will be awarded at most 30% of the maximum points.
 - Each unique compiler warning will result into a deduction of one point.
 - Non-compliance will result into a deduction ranging from one point to five points.
 - A syntax error will result into a score of 0 for the associated challenge.
6. You will be required to submit C source files, header file, and text files and a test script document via Canvas before a specified deadline.
7. **Questions about the MP should be POSTed in the Canvas Discussion Q&A on Machine Problem. Please do NOT email me about the MP except when your reason is a personal one.**

Philippine Stock Market Data: (some simple) Analysis

I. INTRODUCTION

Big Data and *Data Science* are recent buzzwords in multiple disciplines including the sciences and engineering. For this machine problem, we will use real-life data generated from the daily buying/selling of publicly listed stocks in the Philippine Stock Market. More specifically, we will use data for the 30 companies in the Philippine Stock Exchange Index (PSEI)¹ from the years 2015 to 2019. The logos some of these companies are shown in Figure 1. *How many of these logos look familiar to you?*



Note: logo images obtained from PSE website and some of the companies

Figure 1. Logos of some of the companies in the Philippine Stock Exchange

¹ Refer to [The Philippine Stock Exchange, Inc. — PSE](https://www.pse.com.ph/)

Concepts covered in CCPROG2, i.e., arrays, string, structures, text and binary file processing will be applied. Through this MP, you'll need to demonstrate that you can:

- perform data gathering
- design and implement your own data structure for representing, storing, accessing and manipulating stock historical data
- design and implement your own algorithms for the challenge problems described in the MP specification
- specify test cases
- test and debug programs
- properly document and articulate your solution to the MP

Watch the following videos for background information about stocks, stock market and the Philippine Stock Exchange.

- How does the stock market work?
https://youtu.be/p7HKvgRI_Bo
- Explained | The Stock Market
<https://youtu.be/ZCFkWDdmXG8>
- Overview of the PSE
[Overview of the Philippine Stock Exchange](#)

II. CHALLENGES

Challenge #1: Gather Stock Historical Data (SHD) [10 POINTS]

As a preliminary task, we need to gather **stock historical data** (SHD) for stocks in the Philippine Stock Exchange Index (PSEI). This is an easy clerical task that can be done in less than a day since the data that we need are downloadable for free from the Wall Street Journal website. For example, try downloading data for Ayala Corporation from <http://quotes.wsj.com/PH/AC/historical-prices>.

Do this by setting the start date to **1/1/2015** and end date to **12/30/2019**, and then press the “GO” button. Thereafter, press the “DOWNLOAD A SPREADSHEET” button to download the data into an Excel CSV file. A sample screen capture is shown in Figure 2 for your reference. Open the CSV file using Excel or Google Sheets. As shown in Figures 3 and 4, it contains data in six columns, namely: Date, Open, High, Low, Close and Volume². The Open, High, Low, Close are abbreviated as OHLC. Take note of the last row number that contains data. In the AC example, that row number is 1217. Select all cells (press Ctrl + A), copy (press Ctrl + C) and paste (press Ctrl + V) **to a new text file**. Close the Excel file, you will no longer need it.

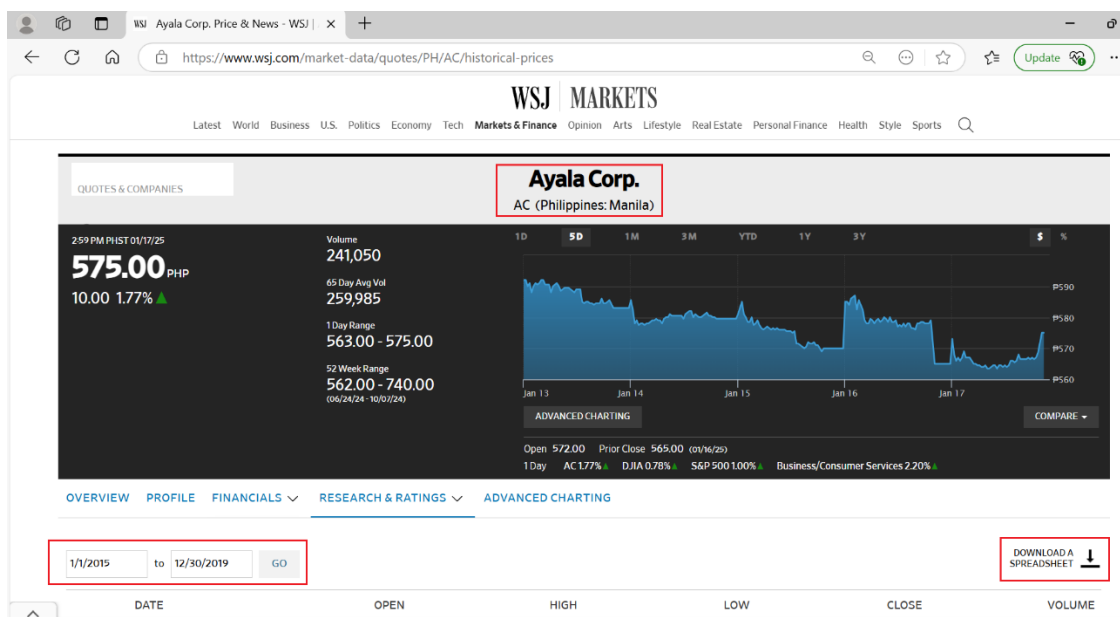


Figure 2. Screenshot for downloading AC stock historical data from 1/1/2015 to 12/30/2019.

² “Open” is the price on the very 1st trade of the day, while “Close” is the price at the end of the day. “High” and “Low” are the highest and lowest prices of the stock respectively for the day. “Volume” is the number of trades.

	A	B	C	D	E	F
1	Date	Open	High	Low	Close	Volume
2	12/27/19	774.941	782.691	760.895	760.895	448810
3	12/26/19	738.131	786.565	738.131	774.941	235550
4	12/23/19	755.083	755.568	744.428	754.599	401280
5	12/20/19	732.319	755.083	721.18	755.083	981550

Figure 3. Screenshot for AC stock historical data.

1214	01/08/2015	679.042	691.635	679.042	687.76	437020
1215	01/07/2015	679.042	683.886	671.293	679.527	290580
1216	01/06/2015	673.23	682.433	672.746	679.042	403100
1217	01/05/2015	676.621	677.105	670.808	675.168	342780
1218						

Figure 4. Row 1217 contains the last row of daily transaction data.

Make sure you're now in the new text file with the stock data; do the following.

- Delete the first row (i.e., the line containing the words Date Open High Low Close and Volume).
- Replace it with a line that contains the stock symbol, followed by at least one space, followed by a number indicating the number of daily transactions. The number of daily transactions is the last row number in the Excel CSV file minus 1. In the AC example, it corresponds to 1216.
- Thereafter, insert one empty line before the first daily transaction data.
- Save the text file using the concatenation of the stock symbol and "orig" as filename. For example, in the case of AC data, the text file should be stored as **ACorig.txt** (Note: you will receive a copy of ACorig.txt file together with this MP specification document; open and examine its contents).

Your 1st Challenge is to gather the historical data for the remaining 28 PSEI stocks. The stock historical data for two companies, specifically, AC and MPI³ are already provided to you. Refer to the accompanying file **PSEI-List2019 . PDF** for the names and stock codes. Just copy and paste <http://quotes.wsj.com/PH/AC/historical-prices> on your browser, then replace AC with an appropriate stock code and proceed as described in the previous paragraphs.

The number of daily transactions is not the same for all stocks! You will need to check the last row that contains data in the Excel file.

DELIVERABLE: store ALL the 30 stock historical data text files on a folder with **GROUPNUMBER-C1** as folder name. Make sure that there are no other unnecessary files! Submit a ZIP file of your folder before the indicated Canvas submission deadline. Make sure to replace GROUPNUMBER with your own 2 DIGIT group number. For example, if your group number is 01, then your ZIP file should be **01-C1 . zip** which when unzipped will produce the folder **01-C1** which in turn contains the 30 text files.

Challenge #2: Representing and Storing the SHD using 1D and 2D Arrays, Input and Output (I/O) Redirection. [10 POINTS]

For this task, you will apply your knowledge of 1D array, 2D array and string processing using the SHD. Write a C program that will read all the contents of one text file (the source file) via input redirection. Do some processing (to comply with the requirements in the two bulleted items below), and produce another text file (the destination file) via output redirection. The source and destination files contain the same information. The difference is in the format of the values, specifically:

- The destination files should contain dates in uniform format following MM/DD/YYYY. For example, if the original data is "1/1/2014", then the date in the destination file should be "01/01/2014". Another example, if the original date is "1/23/18", then the date in the destination file should be "01/23/2018".

³ Metro Pacific Inc (MPI) is already delisted, and its historical data are no longer accessible in the WSJ website.

- The values for OHLC and Volume should be written in the destination file as **double precision floating point** values with two digits after the decimal point.

Hard Requirements: Your program should

1. read via **scanf()** the date entry as a string. It should also write via **printf()** the date as a string.
2. use a 1D array of strings to store the date values, use a 2D array of double for the OHLC values, and a 1D array of double to store the volume values.

Do NOT use **struct** data type and file processing library functions such **fscanf()** and **fprintf()** yet for this challenge.

It is up to you to decide how many white spaces you want to put to separate two data values within the same line of text.

Complete the accompanying skeleton file named **GROUPNUMBER-C2.c**. Compile, run and test your solution exhaustively.

HOW TO RUN and TEST YOUR PROGRAM: You should run the exe file of your program in the command line with **input and output redirection**. For example:

```
C:\CCPROG2> 01-C2 < AOrig.txt > AC.txt
```

where **01-C2.exe** is the student's executable file (for group number 01), **AOrig.txt** is the source file, and **AC.txt** is the destination file. The **<** (less than symbol) is for input redirection, and the **>** (greater than symbol) is for output redirection. **Make sure to test your solution on the accompanying AOrig.txt.**

DELIVERABLE: submit your source code **GROUPNUMBER-C2.c** via Canvas before the indicated deadline.

More challenges in Part 2...