

LAB 7 - TASK 13

Stocks

John Dempsey

COMP-232: Programming Languages
California State University, Channel Islands

October 4, 2023

Hard Due Date: October 11, 2023

Soft Due Date: October 25, 2023

Task 13. Stocks

Want your money to work for you? While this is not financial advice and like any investment strategy does come with risks, one way is to invest in the stock market.

In this assignment, we will write **stocks.c** program to track the performance of four (4) companies and an Index 500 fund.

Here are the ticker symbols we will use:

APPL	Apple Inc.
DIS	Walt Disney Company
FXAIX	Fidelity 500 Index Fund
NIO	NIO Inc.
TSLA	Tesla Inc.

Your program will display the gain/loss for the above symbols for a 10 year period (2013), a 5 year period (2018), or year to date (YTD) by running:

% stocks 2013 ← 10 year period

% stocks 2018 ← 5 year period

% stocks 2023 ← Year to Date

If the user doesn't provide a year or the year isn't 2013, 2018, or 2023, you will need to display the following error messages:

john@oho:~/LAB4\$ **stocks**

Usage:

stocks YYYY

where

YYYY represents 2013 (10 year), 2018 (5 year), or 2023 (YTD).

john@oho:~/LAB4\$ **stocks 1999**

ERROR: Year not 2013, 2018, or 2023.

Usage:

stocks YYYY

where

YYYY represents 2013 (10 year), 2018 (5 year), or 2023 (YTD).

There are three input files: 2013.txt, 2018.txt, and 2023.txt. These files contain the opening stock price on January 1, 2013 (10 years), January 1, 2018 (5 years), and January 1, 2023 (Year To Date (YTD)).

The file format for each file follows:

<symbol><TAB><#shares><TAB><price paid/share><TAB><company>

where:

symbol	Ticker symbol for a company or index fund.
<TAB>	Tab character
#shares	The number of shares purchased on January 1 for the year.
<TAB>	
price paid/share	Opening price on January 1 for the year.
<TAB>	
<company>	The company name

We'll assume we purchased 10 shares of each company and index fund at the opening price on January 1st for the year specified.

Below are the contents for 2013.txt, 2018.txt, and 2023.txt for the symbols APPL, DIS, FXAIX, NIO, and TSLA.

```
root@comp232:/home/LAB7# more 2013.txt 2018.txt 2023.txt
```

```
.....
```

```
2013.txt
```

```
.....
```

```
APPL    10      19.61   Apple Inc.  
DIS     10     51.10   Walt Disney Company  
FXAIX   10     51.78   Fidelity 500 Index Fund  
TSLA    10      2.36   Tesla Inc.
```

```
.....
```

```
2018.txt
```

```
.....
```

```
APPL    10     43.06   Apple Inc.  
DIS     10    111.80   Walt Disney Company  
NIO     10      6.60   NIO Inc.  
FXAIX   10     94.23   Fidelity 500 Index Fund  
TSLA    10     21.37   Tesla Inc.
```

```
.....
```

```
2023.txt
```

```
.....
```

```
APPL    10    124.17   Apple Inc.  
DIS     10     88.97   Walt Disney Company  
NIO     10      9.63   NIO Inc.  
FXAIX   10    132.59   Fidelity 500 Index Fund  
TSLA    10    108.10   Tesla Inc.
```

You can download 2013.txt, 2018.txt, and 2023.txt in the /home/LAB7 directory on comp232.com.

To start writing this program, first read in the above text files into an array defined by the following structure:

```
#define MAX_COMPANIES    10
```

```
struct company {  
    char    symbol[6];  
    int     number_of_shares;  
    float   share_price_paid;
```

```
char    name[50];
} company[MAX_COMPANIES];
```

For the year provided, I would read in and print out the following values to make sure you're reading in the file correctly.

```
john@oho:~/LAB4$ stocks 2023
```

```
Symbol=:APPL:, Number of Shares=10, Price/Share = 124.17, Name=:Apple Inc.:
```

```
Symbol=:DIS:, Number of Shares=10, Price/Share = 88.97, Name=:Walt Disney Company:
```

```
Symbol=:FXAIX:, Number of Shares=10, Price/Share = 132.59, Name=:Fidelity 500 Index Fund:
```

```
Symbol=:NIO:, Number of Shares=10, Price/Share = 9.63, Name=:NIO Inc.:
```

```
Symbol=:TSLA:, Number of Shares=10, Price/Share = 108.10, Name=:Tesla Inc.:
```

```
Number of companies read: 5
```

And use the following Makefile:

```
john@oho:~/LAB4$ cat Makefile
```

```
# Makefile
```

```
SOURCES=stocks.c
```

```
stocks: stocks.c
```

```
    gcc -g stocks.c -o stocks
```

```
    strip stocks
```

```
clean:
```

```
    rm *.o stocks
```

To run the above Makefile, you can type **make** or **make clean**.

After being able to display the company data found in 2013.txt, 2018.txt, and 2023.txt, the next step is to read in the current stock prices for the above companies.

The **prices.txt** file contains the closing price of the stock on March 30, 2023. The format for the prices.txt file is:

```
<symbol><TAB><MM/DD/YYYY><TAB><closing price>
```

where:

symbol	The abbreviated stock identifier for a company
<TAB>	The tab character
MM/DD/YYYY	Date
<TAB>	
999.99	Stock's closing price on 03/07/2023.

The prices.txt file, which you cannot modify, contains the closing prices for ten symbols, including APPL, DIS, FXAIX, NIO, and TSLA, as of 3/7/2023.

```
root@comp232:/home/LAB7# cat prices.txt
APPL      03/07/2023      151.60
BAC       03/07/2023      33.00
CAT       03/07/2023      246.14
DIS       03/07/2023      99.06
FXAIX     03/07/2023      138.64
NIO       03/07/2023      8.97
NVDA     03/07/2023      232.88
QQQ      03/07/2023      296.34
T        03/07/2023      18.71
TSLA     03/07/2023      187.71
```

The next step is to read in the above prices.txt data into an array defined using:

```
#define    MAX_QUOTES    25

struct prices {
    char    symbol[6];
    char    date[11];
```

```
float price;
} prices[MAX_QUOTES];
```

Once you have been able to read in the 2013.txt, 2018.txt, and 2023.txt files and the prices.txt file, we can generate the three reports for 2013, 2018, and 2023. The format for the report is below. Last price used for the report below is for 3/7/2023.

```
john@oho:~/OHO2022/LAB7/STOCKS$ stocks 2023
```

```
Stock Price Gains/Losses From 1/1/2023 to 03/07/2023.
```

Symbol	Shares Owned	Price/Share	Cost Basis	Last Price	Current Value	Total Gain/Loss	Company Name
APPL	10	\$124.17	1241.70	151.60	1516.00	274.30	Apple Inc.
DIS	10	\$ 88.97	889.70	99.06	990.60	100.90	Walt Disney Company
NIO	10	\$ 9.63	96.30	8.97	89.70	-6.60	NIO Inc.
FXAIX	10	\$132.59	1325.90	138.64	1386.40	60.50	Fidelity 500 Index Fund
TSLA	10	\$108.10	1081.00	187.71	1877.10	796.10	Tesla Inc.

Totals:							
		Cost Basis:	4634.60				
		Current Value:	5859.80				
		Actual Gain/Loss:	1225.20				
		Percent Gain/Loss:	+ 26.44%				

The title of the report will specify the date range we are calculating the values for. The above report represents the price gains/losses from 1/1/2023 to 3/7/2023.

Symbol is the company's trading symbol.

The **Shared Owned** is 10 shares for each company and the FXAIX index fund.

Price/Share is the price we paid per share purchased on 1/3/2023.

The **Cost Basis** is how much you paid for the shares. For APPL, we bought 10 shares at \$124.17 per share, so the cost basis is \$1,241.70.

The **Last Price** is the closing price for APPL on 3/7/2023.

The **Current Value** is the number of shares owned multiplied by the Last Price. For APPL, the Current Value is 10 shares x \$151.60 = \$1,516.00.

The **Total Gain/Loss** is the Current Value minus Cost Basis for all shares owned. For APPL, the Total Gain/Loss on 3/7/2023 is $\$1,516.00 - \$1,241.70 = +\$270.30$.

Company Name is the name of the company.

Under Totals, the **Cost Basis** is the sum of the Cost Basis column value for all companies and index fund.

The **Current Value** for the entire account is the sum of the Current Value column value for all companies and index fund.

The **Actual Gain/Loss** is the amount of money you made or lost. You will need to display a plus ('+') sign to represent a positive gain or a minus ('-') sign to represent a negative loss.

The **Percent Gain/Loss** is the percent gain or loss. You will need to display a plus ('+') sign to represent a positive gain or a minus ('-') sign to represent a negative loss.

For the report, you only need to display the dollar sign '\$' for **Price Paid**.

For the report, you do not need to display commas in the dollar amounts.