**Group HW: Problem 1, Stock CGI Interface**

Write a simple interface to the Yahoo stock quote CGI.

(15 points) You should have an HTML page called ***mystocks.html*** with a multi-line text box (i.e., *textarea* element) in which the user can supply stock symbols for one or more companies. Each line will have one stock symbol. The user will then click on the submit button and the page will send the data to a Python CGI program called ***mystocks.cgi***.

(20 points) For each of the stock symbols given, your program must contact the Yahoo finance CGI whose URL is

[http://quote.yahoo.com/d/quotes.csv?**s=GOOG**&**f=sl1d1t1c1ohgvj1pp2owern**&**e=.csv**](http://quote.yahoo.com/d/quotes.csv?s=GOOG&f=sl1d1t1c1ohgvj1pp2owern&e=.csv)

|  |  |
| --- | --- |
| **Name** | **Value** |
| f | sl1d1t1c1ohgvj1pp2owern |
| s | VARIES BY STOCK! |
| e | .csv |

The argument names for this CGI are:

For s, you will need to extract the stock symbols from the user’s data. The return value of the Yahoo CGI for a given stock symbol is a CSV file. We want *Price, Change, Volume,* and *Avg Daily Volume*. If no stocks are given, print out an appropriate error message.

(15 points) Your program should get the above values for each of the specified stock symbols and generate an HTML page that contains a table. Each row of the table will correspond to one stock symbol and the columns of the table will be: *Symbol, Price, Change, Volume, Avg Daily Volume*. Make sure your program works by entering the following symbols in to the text area (each on a separate line)

GOOG   
MSFT   
EBAY   
AMZN

You should get a 4 row table after clicking submit.

(5 point BONUS) If the change is negative, print that row of the table in red font. If the change is positive, print that row of the table in green font. If the change is exactly 0, print that row of the table in black font.

Your solution should look like http://cgi.soic.indiana.edu/~johfdunc/mystocks.html

(without bonus) and should exhibit the same behavior.

**Group HW: Problem 2, Crack the Passphrase!**

Each group has been assigned a unique passphrase. Visit this page:

[**http://cgi.soic.indiana.edu/~johfdunc/password.html**](http://cgi.soic.indiana.edu/~johfdunc/password.html)

The page links to a word list and explains how the passphrases were generated. You’ll need to write code similar to the Safe Cracker to discover your group’s passphrase. When each group guesses their passphrase, it will be recorded here:

<http://cgi.soic.indiana.edu/~johfdunc/pw_guesses.txt>

You’ll need to write a program to load the word list (use a local copy OR the online version) (10 points), search all possible combinations (30 points), and print out a message to alert you to when the correct passphrase is found (10 points).

(10 point BONUS) Find the correct passphrase for your group. You don’t need to submit it – we’ll see it in the log file if your group guesses the passphrase correctly. I don’t recommend trying this search randomly! There are enough words that if you can check 10 words per second, searching the entire space (without duplication) will take a single program up to 1 day.

**Submission**

Upload your programs to Oncourse under Assignments -> Assignment 6 (Group) as .py, .cgi, and .htmlfiles. Name your files ***YourUsername\_A6\_mystocks.cgi, YourUsername\_A6\_mystocks.html,*** and ***YourUsername\_A6\_passphrase.py***

Include the following information as a comment at the top of the file you submit:

 Your name

 Your group number

Make sure you submit a copy of the I211 Homework Team Feedback Form (found on Oncourse). You must submit a copy of this form with every homework or lose 50% of your grade!

**Each member of the group must submit their own copy of the homework!**

Test your code thoroughly!