CS 2204 Lab 10

your name here (please print):

your student ID number here:

Create a subdirectory called `lab10` under your home directory. Perform any necessary work for this lab assignment in that directory. Ensure that this assignment is performed from within a `bash` shell. The goal of this assignment is to write a shell script called `stockquote` that takes a stock market ticker symbol as input and prints the current stock price of the company. Below are sample usages (you may assume that `stockquote` will be called with zero or more legal ticker symbol terms).

```
>stockquote MSFT
The stock of Microsoft Corp. (MSFT) is priced at $25.53.

>stockquote MSFT SUNW
The stock of Microsoft Corp. (MSFT) is priced at $25.53.
The stock of Sun Microsystems Inc. (SUNW) is priced at $3.88.

>stockquote
stockquote: too few arguments.
Usage: stockquote TICKERSYMBOL...
```

Here's how you can approach this problem.

1. First, observe that the URL stem to obtain stock quotes is `http://finance.yahoo.com/q?s=`. You just have to append the ticker symbol to the right of the = sign, e.g., `http://finance.yahoo.com/q?s=MSFT`, to create a well-formed URL. We will concentrate on traditional company stocks (not mutual funds) so that example ticker symbols are `MSFT`, `IBM`, `GOOG`, and `SUNW`. You can then use the `wget` command which takes a URL as input and fetches the webpage. Do a `man wget` to understand how `wget` works. The two options that are important for you are `-o` (to redirect the standard output; `wget` produces a lot of status output) and `-O` (to redirect where you want the fetched page to be stored).

2. Once you have fetched the page and stored it in a file, you must parse this file (e.g., using `sed` or `gawk`) to obtain the stock market price (note that the prices given in the above example might be different from the current price). First inspect the URL in a regular web browser (e.g., Firefox, Mozilla), eyeball the page, and see which portion of the webpage contains the information you need. Then go back to the source (which is the file that you saved via `wget`) and see how to trap that line using an `sed` or `gawk` pattern. If you plan this step well, you can write a search pattern that is independent of the particular ticker symbol.

3. So far, so good. Now you know how to obtain the price. You now have to search for the name of the company corresponding to the ticker symbol. This is a bit more tricky. The name of the company corresponding to ”`MSFT`” is ”`Microsoft Corp.`”. You first have to search for lines where these two appear in proximity to each other (once again, look at the page in a web browser to orient yourself). This time, in using `sed` or `gawk` to extract the name, note that the search pattern will differ for different ticker symbols. This means that the search pattern has to be constructed within your bash script itself.

As an example of how to do this, we give the following sed skeleton below. This is merely a skeleton and is not the correct pattern that you might need to use (you might not even choose to use sed and instead opt for gawk or something else); the skeleton is only meant to illustrate how to construct a ‘dynamic search pattern.’

```
sed -n ’/’$name’/ {
       p
       }’ filename
```

Suppose name is a variable in your bash script that has the value MSFT, the above command inside your bash script is equivalent to saying:

```
sed -n ’/MSFT/ {
       p
       }’ filename
```

i.e., the search pattern is MSFT and the action of the sed command is to print (p) all lines matching the pattern, and the input is the file filename. Once you have constructed a dynamic pattern, run sed or gawk command within your bash script to obtain the company name.

The single quotes around $name essentially ensure that it is evaluated. Without them, the search pattern would actually be /$name/.

4. Now that you have both the company name and the stock price, pretty print the result!

5. Now go back and add if...fi blocks to trap the case when the user does not give any arguments. Use the basename command illustrated in class on Monday.

Here are more hints on how to proceed:

1. Use the -xv option to help debug your script.

2. When using sed, gawk, and wget commands, with complex arguments and parameters, first try these commands on the command line, ensure they are working right, and only then put them inside your bash script. Directly trying to test them from within your bash script will only be a nightmare.