End Sem Test
November 6, 2023

==========**Instructions**==========

1. Submit a zip file as <RollNo>_EndSem.zip containing the codes.

==========

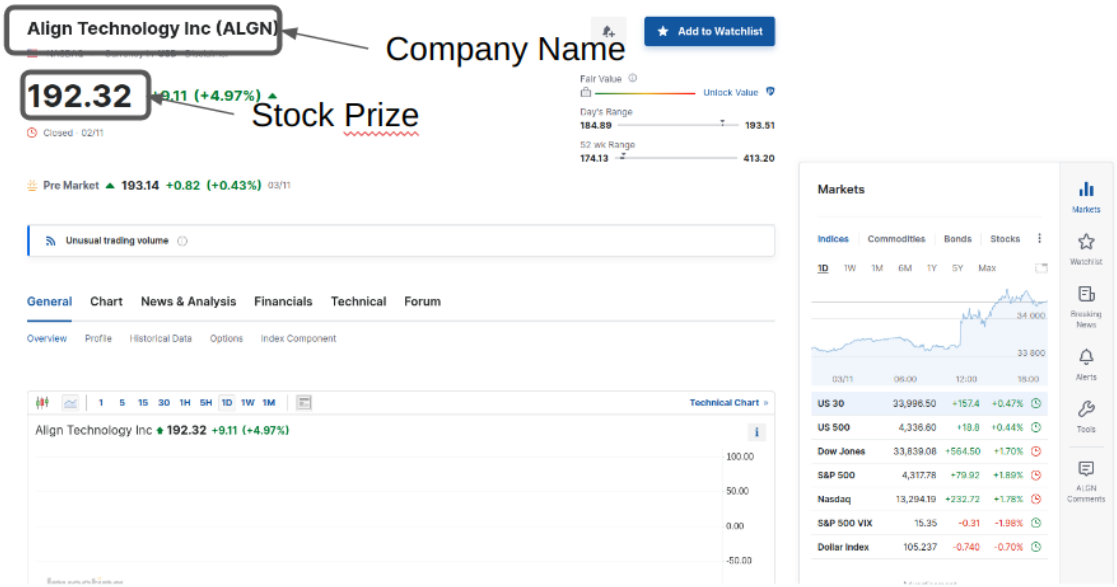
You are a software developer working for a financial company that provides stock market analysis and insights. You are tasked with developing a system that can collect and store a list of stocks from a web source and output the highest performing stock from the collected list.

Your job is mainly comprised of two tasks each of which we discuss below:

Task1: Parsing Stock Pages to Identify Company Name and Stock Price

You have to download the html dump of all the stock market pages from moodle. It is a zip file which contains HTML dump of the stock page for some 30 companies. Unzip it at a location of your preference.

Here is a sample page for your reference:



You have to write a python code (**parser.py**) that does the following:

1. Takes an html file (to be parsed) and a name of file (in which results will be stored) as command line arguments.
2. goes over the html file and scrapes company name and stock price from the html page and stores it in the file passed as command line argument.

Task 2: Identifying the Best Performing Stock in a Multi-threading Setup

In this task, you will find the best performing stock in a multi-threading setting with a shared file. Write a C program that does the following:

1. Find the number and names of files in the html zip that you extracted. [Hint: Use system calls. or you can hard-code it too if nothing works]
2. Create a file (parsed_data.log) which will be a shared file among the threads that will store data returned by the scrapper specific for each html file.
3. Launch as many threads as there are files in the zip. Each thread (specific to a stock market html file) is expected to do the following:
    a. Call the parser.py for the html file with appropriate command line arguments.
    b. Once the parser finishes, read the data from the file in which the parser stored the company name and stock price.
    c. Then, write the information read into the shared file — parsed_data.log. (Ensure proper synchronization to avoid any data loss)
4. Once all the threads have completed parsing their assigned html files, read the shared file —parsed_data.log. Then, get the name of the stock which has the highest price in the file along with the price of the stock. Print the information onto the terminal.

Here is a sketch of how your C code might look like:

```
#include…

void thread_handler(give arguments):
        calls scrapper.py with arguments
        wait for the scrapper to finish (probably use fork()?)
        reads the data parsed from scrapper
        write it into share file –ensure proper synchronization

main():
        find the #files in unzipped html folder/ hard code it
        create a file to be shared among threads
        create a synchronization mechanism for the shared file (can be global!!)
        launch the threads (call thread_handler with appropriate arguments)
        once threads finish, from shared file obtain company names and stock prices
        find max stock price and its associated company
```