**Python Exercises**

Please complete the following exercises and provide your python code in separate files. You may name your python scripts as exercise#.py. The ‘#’ represents the exercise numbers below.

1. Write a script that loops through numbers 1 to 100 and checks odd or even parity. Please print out the word ‘odd’ or ‘even’ next to the number. The number should be printed with a right alignment. The script should however skip numbers in the 50 range and only print out results for 1 to 49 and 60 to 100.

Example:

1. odd
2. even
3. odd

…

100 even

1. Write a script that figures out the number of business days between two dates. Business days should only take into account weekends, not holidays.
2. Write a script that extracts numbers from a string. The string may have more than one set of numbers in it. Example: ‘abc45 def 53xyz’ would be represented as a list of [45,53].
3. Using the attached **orders.txt** file, please write a script that reads the file into a dictionary. Once the file has been read, please prepend the prefix of ‘USEQ:’ to each symbol. E.g. USEQ:MSFT.

Finally, please print the following details only for Buy orders.

OrderId, Side, (updated) Symbol, ExecQty, ExecPrice

1. Please write a script that takes a list of words and returns the longest word. For example: [‘city’,’state’,’zip’,’address’,’name’]. The script needs to figure out which is the longest word and print out the word, as well as the word length.

**Bash Scripting Exercises**

Please use the attached executions.csv to answer the following questions using bash scripting.

1. Write a bash script or command that generates the total count of buys/sells per symbol.

Sample output:

2 Buy GOOG

3 Buy MSFT

…

1. Write a bash script that uses the executions.csv file to generate an output aggregating trades by Date, Symbol, Side and calculating TotalQty and AvgPrice.

Sample output:

Date Symbol Side TotalQty AvgPrice

2017-10-03 GOOG Buy 1100 954.81

2017-10-03 GOOG Sell 1300 954.82

…

**SQL Exercises**

Using the following tables, please provide the SQL that will produce answers to questions below.

Table name: **orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OrderId | CreateTime | Side | Symbol | Qty | Price |
| 1 | 2017-10-03 15:11:01 | Buy | MSFT | 20000 | 74.3 |
| 2 | 2017-10-03 15:11:02 | Sell | MSFT | 100 | 74.2 |
| 3 | 2017-10-03 15:11:03 | Buy | GOOG | 1100 | 954.87 |
| 4 | 2017-10-03 15:11:04 | Sell | GOOG | 1500 | 954.87 |
| 5 | 2017-10-03 15:11:05 | Buy | YHOO | 500 | 67.7 |
| 6 | 2017-10-03 15:11:06 | Buy | SIRI | 12000 | 5.58 |
| 7 | 2017-10-03 15:11:07 | Buy | AAPL | 900 | 154.01 |

Table name: **executions**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **OrderId** | **CreateTime** | **Side** | **Symbol** | **ExecQty** | **ExecPrice** |
| 1 | 1 | 2017-10-03 15:12:01 | Buy | MSFT | 100 | 74.24 |
| 2 | 2 | 2017-10-03 15:12:02 | Sell | MSFT | 100 | 74.24 |
| 3 | 1 | 2017-10-03 15:12:03 | Buy | MSFT | 100 | 74.25 |
| 4 | 1 | 2017-10-03 15:12:04 | Buy | MSFT | 1000 | 74.26 |
| 5 | 3 | 2017-10-03 15:12:05 | Buy | GOOG | 100 | 954.87 |
| 6 | 4 | 2017-10-03 15:12:06 | Sell | GOOG | 100 | 954.87 |
| 7 | 4 | 2017-10-03 15:12:07 | Sell | GOOG | 1200 | 954.87 |
| 8 | 3 | 2017-10-03 15:12:08 | Buy | GOOG | 1000 | 954.87 |
| 9 | 5 | 2017-10-03 15:12:08 | Buy | YHOO | 300 | 67.7 |
| 10 | 6 | 2017-10-03 15:12:08 | Buy | SIRI | 300 | 5.58 |
| 11 | 6 | 2017-10-03 15:12:08 | Buy | SIRI | 9000 | 5.59 |

1. Total Number of orders and executions by Symbol, sorted in ascending order. Output should contain the following columns – Symbol, # of Orders, # of Executions. Please show all orders.
2. Total Number of shares ordered and shares executed per symbol. Output should contain the following columns – Symbol, Total Ordered Qty, Total Executed Qty, Remaining Qty, Status (Filled or Partially Filled), Average Price. Please show totals for all symbols in the orders and executions tables.

1. List of symbols and side (buy/sell) that have more than 1,000 total shares executed by side. Output should contain – Symbol, Side and Total Executed Qty.