1. Task IOS (2021)

Description of the task

The aim of the task is to create a script for the analysis of the record of the system for trading on the stock exchange. The script will filter records and provide statistics as specified by the user.

Simplified introduction to the issue

The stock exchange trades in securities (eg company shares, bonds), commodities (eg oil, cabbage), etc. Each traded item has a unique identifier, the so-called *ticker (*for example, Intel shares have a ticker on the NASDAQ stock exchange INTC, bitcoin may have a ticker assigned to it BTC). The price of items changes over time. A stock trader enters*position,*
either by buying an item and expecting its price to rise in order to sell it at a higher price (so - called *long* position), or that he sells the item and expects its price to fall so that he can then buy it cheaper (so-called *short* position). A trader can also sell an item that he does not currently own (in reality, it works by "borrowing" it from someone who owns it, selling it, and then buying it at a lower price and "returning" it). In our case, we will assume that the trader's system sends orders to the stock exchange to*purchase (*buy) or *sale (*sell) a certain number of units of an item marked with a ticker.

Script interface specification

NAME

- tradelog - stock market log analyzer

USE

- tradelog [-h | --help] [FILTER] [COMMAND] [LOG [LOG2 [...]]

ELECTIONS

- COMMAND can be one of:
  - o list-tick - list of occurring stock exchange symbols, so-called "tickers".
  - o profit - statement of total profit from closed positions.
  - o pos - list of values of currently held positions sorted in descending order by value.
  - o last-price - Listing of the last known price for each ticker.
  - o hist-ord - histogram report of the number of transactions according to the ticker.
  - o graph-pos - statement of the graph of values of held positions according to the ticker.
- FILTER can be a combination of the following:
  - o -and DATETIME - after: only records after this date are considered (without this date).

    DATETIME is a format YYYY-MM-DD HH: MM: SS.

o -b DATETIME - before: only records BEFORE this date (without this date) are considered.

o -t TICKER - only entries corresponding to the given ticker are considered. With multiple occurrences of the switch, the set of all listed ticker is taken.

o -w WIDTH - for a list of graphs, sets their width, ie the length of the longest line to WIDTH. So, WIDTH must be a positive integer. Multiple occurrences of the switch is a faulty start.

• - h and -help print help with a brief description of each command and switch.

Description

1. The script filters records from the stock trading tool. If a command is also given to the script, it executes the command over the filtered records.
2. If the script does not receive a filter or command, it copies the records to standard output.
3. The script can also process records compressed with the gzip tool (if the file name ends .gz).
4. If the script does not receive log files on the command line (LOG, LOG2 ...), Expects records on standard input.
5. If the script is to list, each item is listed on one line and only once. Unless otherwise stated, the order of the lines is given alphabetically according to the ticker. Items must not be repeated.
6. Graphs are plotted using ASCII and rotated to the right. Each line of the histogram indicates a ticker. A positive value or frequency is indicated by the sequence of the grid character#, negative value (u graph-pos) is represented by a sequence of exclamation marks !.

Detailed requirements

1. The script analyzes logs only from the specified files in the given order.
2. The log format is CSV where the delimiter is a semicolon ;. The format is line, each line corresponds to a record of one transaction in the form

```
DATE AND TIME; TICKER; TRANSACTION TYPE; UNIT PRICE; CURRENCY; VOLUME; ID
```

where

o DATE AND TIME are in the format YYYY-MM-DD HH: MM: SS

o TICKER is a string that does not contain whitespace and a semicolon

o TYPE OF TRANSACTION takes on value buy or sell marking purchase resp. sale

o UNIT PRICE is the price per share, unit of commodity, etc. to the nearest a maximum of two decimal places; the dot character serves as a separator of units and decimals.; E.g.1234567.89

o CURRENCY is a three-letter currency code, e.g. USD, EUR, CZK, SEK, GBP etc.

o VOLUME indicates the number of units (stocks, commodity units, etc.) in a transaction

o ID is the transaction identifier (a string without whitespace and a semicolon)

Transaction value Yippee UNIT PRICE * VOLUME. Example records:

```
2021-07-29 23: 43: 13; TSM; buy; 667.90; USD; 306; 65fb53f6-7943-11eb-80cb-8c85906a186d 2021-07-29 23:
43: 15; BTC; sell; 50100; USD; 5 ; 65467d26-7943-11eb-80cb-8c85906a186d
```

- The first record marks the purchase of 306 shares of TSMC (ticker TSM) for the price of 667.90 USD / stocks. The value of the transaction is therefore 204377.40 USD.
- The second record indicates the sale of 5 bitcoins (ticker BTC) for the price of 50,100 USD / bitcoin. The value of the transaction is therefore 250500.00 USD.

3. Assume that the currency is the same for all records (no need to verify).
4. The script does not modify any file. The script does not use temporary files.
5. You can assume that the entries in the input files are listed chronologically, and if there are multiple files in the input, their order is also chronological.
6. *Total profit on closed positions (*command profit) is calculated as *sum of values sell transactions - the sum of values buy transactions.*
7. *Value of positions currently held (*orders pos and graph-pos) is calculated for each ticker as *number of units held * unit price from the last transaction,* where *number of units held* is given as *sum of volumes buy transactions - the sum of volumes sell transactions.*
8. If not when using the command hist-ord indicated width WIDTH, then each position in the histogram corresponds to one transaction.
9. If not when using the command graph-pos indicated width WIDTH, then each position in the histogram corresponds to a value of 1000 (rounded to the nearest thousand, ie the value 2000 will be represented as ## while the value of 1999.99 as # and value - 1999.99 as !.
10. For commands hist-ord and graph-pos with the specified width WIDTH when dividing, round to zero (ie for graph-pos -w 6 and the longest line with a value of 1234, the line with a value of 1234 will look like this ######, line with the value 1233.99 as follows ##### and a line with a value of -1233.99 as follows !!!!!).
11. The order of arguments is enough to consider such that first there will be all the switches, then (optionally) the command and finally the list of input files (so you can use getopts). Supporting arguments in any order is an optional extension, the implementation of which can compensate for the possible loss of points in another part of the project.
12. Assume that input files cannot have names that correspond to any command or switch.
13. In case of putting a switch -h or -help only the help is always written and the script ends (that is, if the switch is followed by a command or file, it will not be executed).

14. When listing using commands pos, last-price, hist-ord and graph-pos the tikers must be left-aligned and the colon in the 11th position on the line (fill in with spaces). For commandshist-ord and graph-pos there is exactly one space after the colon on all lines (or none if there is nothing in the right column of the given line), for commands pos and last-price the values in the right part of the report are formatted so that (in the case of a non-empty report) there is exactly one space on the line with the longest string representation of the value (ie including the sign) between the colon and the value and the other lines are right-aligned with examples of statements below).

Return value

- The script returns success if the operation is successful. An internal script error or incorrect arguments will be accompanied by an error message and a failed return code.

Implementation details

1. The script should be set up at runtime `POSIXLY_CORRECT = yes`.
2. The script should run on all common shells (`dash`, `ksh`, `bash`). If you use a shell-specific property, specify it using the interpreter directive on the first line of the file, e.g. `#! /bin / bash` or `#! /usr / bin / env bash` for `bash`. You can use the GNU extension for `sed` whose `awk`. Perl, Python, Ruby, etc. are not allowed.

   NOTICE: some servers, e.g. `merlin.fit.vutbr.cz`, have a symlink `/bin / sh -> bash`. Therefore, make sure that you are actually testing the script with that shell. I recommend verifying the correct functionality using the virtual machine below.

3. The script must run on the commonly available GNU / Linux, BSD and MacOS OS. A virtual machine with a downloadable image is available to students here: http://www.fit.vutbr.cz/~lengal/public/trusty.ova(for VirtualBox, login: `trusty` / password: `trusty`), on which the correct functionality of the project can be verified.
4. The script must not use temporary files. However, temporary files indirectly created by other commands (such as`sed -i`).
5. Write the numbers in decimal notation to two decimal places. Attention, some tools (eg`awk`) they can list larger numbers implicitly using scientific notation.

Project submission

Submit only the script `tradelog` (do not pack it in any archive). Submit to the IS, deadline Project 1.

Advice

- A good decomposition of the problem into subproblems can make your work much easier and prevent mistakes.
- Learn *good* use functions in the shell

Examples of use

- Samples of stock exchange instrument entries are available here: https://pajda.fit.vutbr.cz/ios/ios-21-1-logs

Examples:

```
$ cat stock-2.log | head -n 5 | ./tradelog
2021-07-29 15: 30: 42; MSFT; sell; 240.07; USD; 327; 65fad854-7943-11eb-929d-8c85906a186d 2021-07-29 15: 31:
12; MA; sell; 314.91; USD; 712 ; 65fae24a-7943-11eb-9171-8c85906a186d 2021-07-29 15: 31: 32; BAC; buy; 34.16;
USD; 635; 65fae466-7943-11eb-8f48-8c85906a186d 2021-07-29 15:37:09 ; BAC; sell; 36.67; USD; 897;
65fae614-7943-11eb-9ccb-8c85906a186d 2021-07-29 15: 43: 02; JPM; sell; 146.77; USD; 190;
65fae79a-7943-11eb-8977-8c85906a186d

$ ./tradelog -t TSLA -t V stock-2.log
2021-07-29 17: 06: 57; TSLA; buy; 757.57; USD; 812; 65fafb04-7943-11eb-8d41-8c85906a186d 2021-07-29 17: 58: 18; V; sell; 215.31; USD; 406 ;
65fb0662-7943-11eb-87fe-8c85906a186d 2021-07-29 18: 12: 27; TSLA; sell; 729.75; USD; 482; 65fb0892-7943-11eb-867f-8c85906a186d 2021-07-29 18:55:19 ; V;
sell; 217.92; USD; 210; 65fb1238-7943-11eb-86e2-8c85906a186d 2021-07-29 19: 19: 26; TSLA; sell; 700.75; USD; 457; 65fb1792-7943-11eb-8abf-8c85906a186d
2021-07-29 19: 27: 39; TSLA; buy; 710.79; USD; 633;   65fb19b8-7943-11eb-a5d9-8c85906a186d 2021-07-29 20: 06: 53; V; sell; 218.72; USD; 272 ;
65fb237c-7943-11eb-83a3-8c85906a186d 2021-07-29 20: 59: 16; V; sell; 196.54; USD; 92; 65fb2c32-7943-11eb-9dd3-8c85906a186d 2021-07-29 21:03:15 ; V; buy;
188.60; USD; 605; 65fb2d4a-7943-11eb-8804-8c85906a186d 2021-07-29 21: 17: 37; V; sell; 222.52; USD; 447; 65fb2f7a-7943-11eb-8f28-8c85906a186d 2021-07-29
21: 18: 18; TSLA; buy; 733.96; USD; 720;65fb3092-7943-11eb-992a-8c85906a186d 2021-07-29 21: 50: 25; V; sell; 212.58; USD; 2833;
65fb3a2e-7943-11eb-8e0b-8c85906a186d 2021-07-29 22:10:55; TSLA; sell; 718.31; USD; 3794; 65fb3f88-7943-11eb-a371-8c85906a186d 2021-07-29 22: 21: 31;
TSLA; sell; 681.74; USD; 7122; 65fb41a4-7943-11eb-a09f-8c85906a186d 2021 -07-29 23: 01: 47; TSLA; sell; 707.03; USD; 1578;
65fb4a50-7943-11eb-9f6e-8c85906a186d 2021-07-29 23: 21: 11; TSLA; buy; 679.27; USD; 9655; 65fb4fb4-7943-11eb-8199-8c85906a186d 2021-07-29 23: 43: 13;
TSLA; buy; 667.90; USD; 306; 65fb53f6-7943-11eb-80cb-8c85906a186d 2021-07-29 23:48:29; V; buy; 195.52; USD; 2003;
65fb5824-7943-11eb-9b59-8c85906a186d65fb41a4-7943-11eb-a09f-8c85906a186d 2021-07-29 23: 01: 47; TSLA; sell; 707.03; USD; 1578;
65fb4a50-7943-11eb-9f6e-8c85906a186d 2021-07-29 23:21:11; TSLA; buy; 679.27; USD; 9655; 65fb4fb4-7943-11eb-8199-8c85906a186d 2021-07-29 23: 43: 13;
TSLA; buy; 667.90; USD; 306; 65fb53f6-7943-11eb-80cb-8c85906a186d 2021 -07-29 23: 48: 29; V; buy; 195.52; USD; 2003;
65fb5824-7943-11eb-9b59-8c85906a186d65fb41a4-7943-11eb-a09f-8c85906a186d 2021-07-29 23: 01: 47; TSLA; sell; 707.03; USD; 1578;
65fb4a50-7943-11eb-9f6e-8c85906a186d 2021-07-29 23:21:11; TSLA; buy; 679.27; USD; 9655; 65fb4fb4-7943-11eb-8199-8c85906a186d 2021-07-29 23: 43: 13;
TSLA; buy; 667.90; USD; 306; 65fb53f6-7943-11eb-80cb-8c85906a186d 2021 -07-29 23: 48: 29; V; buy; 195.52; USD; 2003;
65fb5824-7943-11eb-9b59-8c85906a186d

$ ./tradelog -t CVX stock-4.log.gz | head -n 3
2021-09-27 05: 12: 30; CVX; sell; 108.17; USD; 88; 8f229a62-7945-11eb-a6fb-8c85906a186d 2021-09-27 13: 57:
48; CVX; sell; 94.81; USD; 5374 ; 8f22ec38-7945-11eb-8c68-8c85906a186d 2021-09-27 14: 52: 50; CVX; sell;
89.22; USD; 7759; 8f22f46c-7945-11eb-9bb2-8c85906a186d

$ ./tradelog list-tick stock-2.log AAPL

AMZN
BABA
BAC
DIS
FB
GOOG
GOOGL
JNJ
JPM
MA
MSFT
NVDA
PG
PYPL
TSLA
TSM
UNH
IN
WMT

$ ./tradelog profit stock-2.log
- 58863165.03

$ ./tradelog -t TSM -t PYPL profit stock-2.log
- 577302.62
```

```
$ ./tradelog pos stock-2.log AMZN:
64645275.64
GOOGL: 7914389.08
NVDA: 2540507.69
DIS: 1925621.88
TSM: 1266217.38
JPM: 937220.31
BABA: 444692.64
BAC: 323899.29
JNJ: 81769.32
FB: 42673.05
WMT: 2423.34
MSFT: -321051.64
V: -322999.04
PYPL: -502892.46
MA: -569746.42
TSLA: -872945.30
PG: -1138885.10
AAPL: -1190996.48
UNH: -1781240.88
GOOG: -9846258.51

$ ./tradelog -t TSM -t PYPL -t AAPL pos stock-2.log TSM: 1266217.38

PYPL: -502892.46
AAPL: -1190996.48

$ ./tradelog last-price stock-2.log AAPL: 133.88

AMZN: 3496.04
BABA: 245.28
BAC: 38.61
DIS: 207.48
FB: 275.31
GOOG: 1975.97
GOOGL: 1990.04
JNJ: 155.16
JPM: 135.77
MA: 333.38
MSFT: 237.64
NVDA: 629.93
PG: 124.70
PYPL: 279.54
TSLA: 667.90
TSM: 140.41
UNH: 321.06
V: 195.52
WMT: 134.63
```

```
$ ./tradelog hist-ord stock-2.log AAPL: ##

AMZN: #####
BABA: ####
BAC: #####
DIS: #####
FB: ####
GOOG: ######
GOOGL: ########
JNJ: ##
JPM: ######
MA: ####
MSFT: ####
NVDA: ######
PG: #####
PYPL: ####
TSLA: #########
TSM: ##
UNH: ######
V: ########
WMT: ####

$ ./tradelog -w 100 graph-pos stock-6.log AAPL: !!!!!!!!!!!!

AMZN: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!
BABA: ####
BAC: ###
DIS: ##################
FB:
GOOG:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!!!

GOOGL       : ############################################ #############################:
JNJ
JPM         : #########
MA          : !!!!!
MSFT        : !!!
NVDA        :
PG          ########################: !!!!!!!!!!!
PYPL        : !!!!!
TSLA        : !!!!!!!!
TSM         : ###########
UNH         : !!!!!!!!!!!!!!!!!!
IN          : !!!
WMT         :

$ ./tradelog -w 10 -t FB -t JNJ -t WMT graph-pos stock-6.log FB: #####

JNJ: #########
WMT:

$ cat / dev / null | ./tradelog profit
0.00
```