CLI APP COURSE (cryptotocker)

https://en.wikipedia.org/wiki/Command-line_interface#Anatomy_of_a_shell_CLI

Helpful links:
https://cplusplus.com/articles/DEN36Up4/

https://www.youtube.com/watch?v=1DR-Sj-RLzQ
https://www.youtube.com/watch?v=MBqo7Bw_R5o

Api to fetch data from:
GET https://api2.binance.com/api/v3/ticker/24hr

Useful packages (not an exhaustive list):
https://github.com/friedmud/variadic_table
https://github.com/elnormous/HTTPRequest
https://github.com/nlohmann/json
https://github.com/aafulei/color-console

Linear regression testing tool online:
https://www.graphpad.com/quickcalcs/linear1/

App should display the info for the current state of the crypto symbols in the api according to the input parameters.

Output should be formatted like this (using the table package):

Last updated: 2022-09-01 13:05:33 (x seconds ago)

```
-------------------------------------------------------------------------------
| symbol    |    price     |    price change   |  volume  | number of trades |  trend  |
-------------------------------------------------------------------------------
| BTCUSDT | 20000.00 |         5.5%          | 4.52 B  |          32313          |   52.3   |
.......
```

Volume column should be formatted using abbreviation. Ex: https://stackoverflow.com/a/41621940/1270586
Last updated timestamp should be formatted from the timestamp when data get request completed.

Table columns:

- symbol
- price (last available price for symbol)
- price change (displayed in percentages ex: "5.2%"; "priceChangePercent" in json)
  - text should be green if the percentage is positive and red if is negative
- volume ("quoteVolume" in json)
- number of trades ("lastId" in json)
- trend
  - a linear regression calculation performed on the price column for a given number of continuous (no gaps) samples from past
  - column is displayed only if the trend flags is present
  - value range: -100 ~ 100 (from line goes straight down, to line straight up);
  - text should be coloured with gradient from green (100) to white (0) to red(-100), according to the value displayed
  - trend calculation should persist even if the application is restarted

+ bonus points for extra features

---

flags (i.e. command line arguments for the program)

- --sort
  - sorts table values according to the arguments
  - given a pair of column name + "asc" or "desc"
  - should handle multiple column sorts if multiple sort args are passed
  - ex usage: --sort price asc
  - ex usage: --sort price asc --sort volume desc
  - ex usage: --sort price asc volume desc
  - should return error if param is invalid
- --limit
  - limits the number of rows displayed in the table
  - is applied after the sort flag
  - receives a positive integer number as argument
  - should return error if the input is invalid
  - ex usage: --limit 10
- --pairs
  - filters what symbols should be displayed
  - if the pair flag is missing, all pairs are displayed
  - input is a space separated list of symbols
  - symbol names should be case insensitive

    - should support wildcard (*)
    - ex usage: --pairs BTCUSDT BTCusdt BTC*sDt (all these params are equivalent and match to BTCUSDT)
- --ignore-errors
    - a flag that if it exists, no errors are displayed
    - if the program has no errors it works as expected
    - if the program encounters errors, stops executing and returns as if it was successful
    - ex usage: --ignore-errors
- --live
    - if the flag exists, the table above refreshes the data
    - supports optional integer parameter that represents the refresh interval (in seconds)
    - if the live flag is not present, it displays the data once and the program terminates
    - if the refresh interval parameter is missing it defaults to 5 (seconds)
    - ex usage: --live 10 (refreshes the table every 10 seconds)
    - ex usage: --live (refreshes the table every 5 seconds)
- --trend
    - a flag for the proce trend
    - inputs a integer for the number of samples on which the linear regression is calculated
    - the input number is the number of 'ticks' of the refresh param
    - example: if the live flag is present, and it has a 10 seconds refresh period and the trend number is 6 the trend is computed on the last 60 seconds (10 seconds per refresh with 6 samples required)

Executable name should be 'cryptoticker'.

Given the name of the executable is 'cryptoticker' some example usages are:

cryptoticker --sort price asc --sort symbol desc —limit 10 —pairs BTCUSDT ET*USDT --ignore-errors --live --trend 5

cryptoticker --sort price asc --sort symbol desc —limit 10 —pairs BTCUSD* --live 5

cryptoticker --sort trend desc --pairs BTCUSDT --live 5

cryptoticker --live