Assignment 1a

You are required to write a software in **C++** that is able to subscribe to websocket connections asynchronously, as well as polling for information via HTTP REST connection periodically and asynchronously using the links below:

https://binance-docs.github.io/apidocs/futures/en/#order-book
https://binance-docs.github.io/apidocs/futures/en/#partial-book-depth-streams

You may choose to connect to any symbol of your choice. For example, BTCUSDT and ETHUSDT.

For both of the requirements above, ensure that you:

1. Connect to multiple sessions and manage the threads efficiently.
2. Use relevant queue and/or buffer mechanism, if any, to manage the inbound traffic and tasks.
3. Deduplicate all the messages so that each callback is unique.
4. Manage the resources (CPU) efficiently. A CPU is required to perform both REST connectivity and websocket connectivity. You may split symbols horizontally.
5. Trigger callbacks for the messages with these function names: OnOrderbookWs, OnOrderbookRest

You may use readily available packages such as websocketpp client. However, extra attention will be given to candidates who put in the effort to present a more elegant and low latency solution. You are strongly encouraged to use lower level libraries such as boost. You may use CMakeLists to manage your project structure.

Submit your code and ensure that it can be compiled and run.

Assignment 1b

Answer the following questions in words and diagrams:

1. For the solution above, explain in a diagram how each CPU changes context with relevance to your threading mechanism and task schedulers, where the x-axis represents the task and the y-axis represents time. Show clearly how the websocket and REST connectivity can co-exist without blocking the main loop.
2. What are some of the potential bottlenecks for the system above? Therefore, what should be monitored?
3. Given more time, what can we do to improve latency for the task above?