**Purpose:** Create a client-server application for Windows.

**Algorithm:**

* Clients send to the server random integer numbers from 0 to 1023.
* Server receives and stores these numbers into a container as unique values.
* Upon a client's request, the average of squares of the numbers including the newly received number is calculated and sent back to the client as a response.
* A client receives this value, generates a new random number and then again sends it to the server and so on.
* Every N seconds the server makes a dump - it stores all numbers from the container into a file in the binary format

**Details:**

* User has to be able to normally stop clients and server (for example, using ESC button)
* Server has to make a dump in the separate thread - not in the thread(s) which is used for communication with clients
* Output all the processes into console and the application's log file
* **When developing, you must use the Boost Asio library**

**Notes:**

* Consider to develop this task as a production-like code. **We will pay attention not only on code correctness but also on code quality**
* For tests purposes run 1 server and about 10 clients. Keep them working for 10-20 minutes.

**Цель:** Создание клиент-серверного приложения под Windows.

**Алгоритм:**

* Клиент посылает на сервер запрос, содержащий случайное целое число в диапазоне от 0 до 1023 включительно.
* Сервер получает эти числа и сохраняет их в некий контейнер. Сохраняются только уникальные значения.
* После сохранения числа сервер вычисляет среднее арифметическое от квадратов всех сохраненных чисел и посылает это среднее назад клиенту в качестве ответа на запрос.
* Клиент получает значение, генерирует новое случайное целое число из указанного выше диапазона и снова посылает его на сервер. Эти шаги повторяются вновь и вновь, пока сервер и клиент работают.
* Каждые N секунд сервер должен сделать дамп - он должен сохранить в файл все числа, которые в данный момент хранятся в контейнере. Файл должен иметь бинарный формат.

**Подробности:**

* У пользователя должна быть возможность штатно остановить клиентов и сервер (например, нажатием на ESC). Обычное закрытие консоли не считается штатным закрытием.
* Сервер должен сохранять дамп в отдельном потоке, то есть не в том потоке (или потоках), в котором происходит общение с клиентами
* Сервер и клиент должны писать логи о происходящих событиях в консоль и в свои файловые логи
* **В процессе разработки используйте библиотеку Boost Asio**

**Дополнительная информация:**

* Относитесь к реализации данного тестового задания как к написанию production кода. **Мы будем обращать внимание не только на работоспособность кода, но и на его качество**
* Для тестирования запустите 1 сервер и около 10 клиентов. Дайте им поработать 10-20 минут.