6 Further Java (ARB)

Your task is to implement a trusted timestamp service. The service should accept TCP connections from clients on port 2510, read in ASCII characters from clients up until the first newline character, cryptographically sign and date the characters read, and return the digital signature to the client before closing the connection. You may assume the existence of a class `Util` with a static method `String sign(String document)` which, given a string `document`, will generate a digital signature suitable for sending to the client.

(*a*) Write a single-threaded implementation of the trusted timestamp service. You may use any features of the Java standard library. [6 marks]

(*b*) Write an implementation of a queue as a class `ConcurrentBlockingQueue<T>`, with public methods `void put(T)` and `T get()`. Your implementation should be thread-safe and calls to `get` should block until an item is available for return to the caller. You should not use any features of the Java standard library except for `java.util.ArrayList`. [6 marks]

(*c*) Write a multi-threaded implementation of the trusted timestamp service which uses a static set of ten worker threads, created when the program starts, to service requests from clients. You may use any features of the Java standard library and you may assume an implementation of `ConcurrentBlockingQueue<T>` described in part (*b*). [5 marks]

(*d*) When will your multi-threaded implementation achieve lower-latency responses to clients than the single-threaded implementation? Why? [3 marks]