ECE 522 Assignment 8

Zhaoyi Wang 1689747

Question 1

Question a)

For code: `let addr = ([127, 0, 0, 1], 3000).into();`

The `[127,0,0,1]` means the IP address `127.0.0.1`, and the `3000` means the port. That is, `127.0.0.1:3000`.

Question b)

For code `fn svc_wait(t: u64) -> impl Future {}`

`Future` is a trait under `hyper::rt`.

Futures are used to provide a sentinel through which a value can be referenced. They crucially allow chaining and composing operations through consumption which allows expressing entire trees of computation as one sentinel value.

Question c)

For the website `http://httpbin.org/`, it is a simple HTTP Request and Response Service.

Question d)

`body` at line 45 represents the data from the `httpbin.org`. This part of code is trying to do the deserialization.

Question e)

`BoxFut` type is wrapping the `Future` type with a `Box` type. It is an owned dynamically typed `Future` for use in cases where you can't statically type your result or need to add some indirection.

Question f)

No, it shouldn't.

Question g)

There is NO need to use a lifetime specifier.

Question h)

curl, which stands for *client URL*, is a command line tool that developers use to transfer data to and from a server.

It means that call the sever on one of the endpoint `wait`.

Question i)

This code doesn't use `Async/IO`. If we want to use `Async`, we need to combine `hyper` and `tokio`. Also, we need to use some interface like `AsyncRead` and `AsyncWrite` under `tokio::io`.

Question 2

Question a)

Diem, formerly known as **Libra**, is written in `Move` Language.

Question b)

Move is a safe and flexible programming language for the Diem Blockchain. Move is an executable bytecode language used to implement custom transactions and smart contracts.

> The key feature of Move is the ability to define custom *resource types* with semantics inspired by linear logic: a resource can never be copied or implicitly discarded, only moved between program storage locations. These safety guarantees are enforced statically by Move’s type system. Despite these special protections, resources are ordinary program values — they can be stored in data structures, passed as arguments to procedures, and so on. First-class resources are a very general concept that programmers can use not only to implement safe digital assets but also to write correct business logic for wrapping assets and enforcing access control policies. The safety and expressivity of Move have enabled us to implement significant parts of the Diem protocol in Move, including Diem coin, transaction processing, and validator management. - From Diem Docs.

Question c)

- `lazy-static`
  - Using this macro, it is possible to have `static` that require code to be executed at runtime in order to be initialized.
- `tokio`
  - `tokio` is an event-driven, non-blocking I/O platform for writing asynchronous applications with the Rust programming language.
- `failure`
  - An experimental new error-handling library. `failure` is designed to make it easier to manage errors in Rust. It is intended to replace error management based on `std::error::Error` with a new system based on lessons learned over the past several years, including those learned from experience with quick-error and error-chain.