The goal of this assessment is to showcase your ingenuity and authenticity to craft and solve technical problems while applying Golang syntax and features.

You'll see that the tasks outlined below build on the questions asked in the technical interview. This approach is taken to evaluate your practical response to those questions.

Instructions

1. Ensure the problems you create are relevant to your work experience.
2. Ensure your solutions are well documented to explain your thought process.
3. Prioritize clean, readable, and efficient code.
4. You may use external packages in your solutions.
5. Your work must be submitted as a single Golang CLI application where commands are exposed to execute the code for each task.
6. Submit your code by linking a Github repository (or similar)

Tasks

Task #1: implementation of the “adapter” structural design pattern

- Outline a problem where you needed incompatible `structs` to collaborate in order to provide a common feature-set to the rest of your application.
- Implement a solution.
- Provide commentary on how your implementation can be improved.
- Include tests to verify that your solution works.

Task #2: implementation of concurrency and parallelism

- Outline a problem where the use of concurrency primitives (goroutines and channels) were a suitable solution.
- Implement a solution, ensuring that it handles synchronization and avoids race conditions.
- Provide commentary on how your implementation can be improved.
- Include tests to verify that your solution works correctly under concurrent usage.

Task #3 (Optional): open-ended challenge

Propose and solve a programming challenge that you believe showcases your ingenuity and deep understanding of Go. This could involve algorithms, system design, tool development, or any area you excel in.