Design and implement the **transaction_processor** program in Go programming language. Ensure implementation is production grade. Consider the usual concerns when writing production code, including appropriate tests, maintainability, and performance.

The **transaction_processor** program is responsible for processing a time series JSON feed. Refer to the **data.json** file for the schema and records.

The program should read the records from the *data.json* feed sequentially and ingest them into a Ring Buffer. The Ring Buffer is a capacity-bounded circular queue. It would be ideal if you can make the size of the Ring Buffer configurable.

The Reader of the Ring Buffer should read and process the records independently. For now, the processing step simply involves printing the record to stdout.

Question 2 (Optional):

As an extension to the solution, please add support for multiple readers. This means that the program should allow multiple Reader instances to consume and process the records from the Ring Buffer concurrently. How will you distribute the workload among the multiple readers?

Think about the overall design of the program. How will the components interact with each other? What is the best way to structure the code for maintainability and extensibility?
Think about potential performance bottlenecks in the program. Are there any optimizations you can make to improve processing speed or reduce resource usage?

Write appropriate tests to ensure the correctness of your implementation.