Complete the task

a) using the independent implementation of the synchronizer,

b) from the standard library

c) in the Golang language

a) Tasks about readers and writers. The database is shared by two types of processes - readers and writers. Readers perform transactions that view database records, writer transactions and view and modify records.

Create a multithreaded application that works with a shared file.

To protect operations with a shared file, use a read-write lock. The file contains a sequence of records of the form: NAME 1 - phone 1, NAME 2 - phone 2 ... The following streams should work:

1) flows that find phones by the specified last name;

2) flows to which P.I.B. by the specified phone number;

3) streams that delete and add records to the file.

b) Question about the garden. Create a multithreaded application that works with a generic two-dimensional array. To protect operations with a common array, use a read-write lock. A two-dimensional array describes a garden. The following threads should work in the application:

1) stream-gardener watches over the garden and waters the wilted plants;

2) flow-nature can arbitrarily change the state of plants;

3) stream-monitor1 periodically outputs the state of the garden to a file (without erasing the previous state);

4) stream-monitor2 displays the state of the garden on the screen.

c) Question about the bus. Create a multi-threaded application that works with a general graph.

To protect operations with the graph, use a read-write lock.

The graph describes the set of cities and the set of bus routes from city A to city B with the ticket price (by default, if the flight is from A to B, it also goes from B to A, with the same price). The following threads should work in the application:

1) a flow that changes the ticket price;

2) a flow that removes and adds flights between cities;

3) a stream that removes old cities and adds new ones;

4) flows that determine whether there is a path from an arbitrary city A to an arbitrary city B, and what is the price of such a trip (if there is no direct path, then find any existing path)