Complete the task

a) using an independent implementation of the synchronizer,

b) from the standard library

c) in the Golang language

a) Tasks about new recruits. Many recruits are given the command "to the left"

or "to the right". All recruits try to carry out the order, but the problem is

because they don't know where the right is and where the left is. Therefore, each recruit turns either to the right or to the left. If the new recruit turns around and sees that his neighbor is standing with his back to him, he believes that he has done everything right. If they come face to face, then both believe that they were mistaken and turn 180 degrees. Create a multi-threaded application that simulates the behavior of multiple recruits until it reaches a steady state. The number of recruits ≥ 100. A separate stream is responsible for a part of the order of at least 50 recruits.

b) Create an application with four threads. Every thread is running

with its own line. Strings can contain only A, B, C, D characters.

A thread can change the character A to C or C to A or B to D or D to B. Threads stop when the total number of A and B characters becomes equal for at least three lines.

c) Create an application with three threads. Each thread works with its own array, the threads check the sum of the elements of their array with the sums of the elements of other threads and stop when all three sums are equal. If the sums are not equal, each thread adds one to one element of the array or subtracts one from one element of the array, then retests the sums for equality. At the moment of stopping all three streams, the sums of array elements must be the same.