1DV512 - Lab assignment 2
Student: Marah Awad

- Task 1:
  In the first task, I created two semaphores semA and semB. Each of them has a loop for 10 times in the run-method, because we should print A and B ten times. I start semA with value 1, so it gets the first turn to run and print "A", semA decreases by "acquire()" its value by 1, prints "A" and then gives the turn to semB by "release()" to then do the same thing.

- Task 2:
  In this task, I used the same principle as task 1. I added 2 more semaphores: semC and semD. I start by semA with value 1, to give it a turn to run first. SemA makes "release" -increase with 1- to semB and the next time for semC. I used "i % 2 == 0" to specify which semaphore I will release. When "i" is even >> semB release, and when "i" is odd >> semC release. And same principle for in semC body.

- Task 3:
  This task was different, I created three senders that send A, B, C each separately to the circular queue. An interface IMessageQueue that is implemented by the CircularQ class. In addition to that, I created a Receiver class that tries to receive messages and print them. Most of the work is on the CircularQ class, it has a size, the integer front and rear that represent the index of the first and last elements in the queue. I created three semaphores in it, one of them to control sending messages, one to control receiving messages, and one to protect shared data from being preempted. When a circularQ is created in main, its list starts with zero elements which are represented by front and rear are equal to -1. I added the body for send method that checks if the queue is empty or not when we get access to send and when the queue has available places -more details in code comments-. For the receive-method body, it returns a char/message to the receiver to be printed on the screen. The main role of accessQueue semaphore is to control if a thread/sender is not finished with its sending so it blocks the receiver from receiving other senders' messages until it's done.
  Note: I used sleep time in the sender's body to get better variety of results.