Ece Karasu 29467

**PA3 Report**

I implemented a line (a list of struct) in my code to achieve synchronization. When a thread comes, it adds itself to the line and checks whether a valid combination has occurred.

On the other hand, I used 3 different mutexes for 3 different operations (printing messages, checking for a combination, and adding to the list). That is, for each operation the mutex is locked before executing the operation and unlocked when the operation finishes. This implementation disables multiple threads to execute the same operations at the same time.

Here is the detailed pseudo code of my thread function:

```
is_driver_found = false

wait until the captain finishes its job
lock print_mutex
   print init message
unlock print_mutex

lock add_mutex
   Add the current thread to the list
unlock add_mutex

lock check_mutex
Find the number of A's and B's
If the combination is found
      set is_driver_found =true
      print mid message
      print captain message
else
      wait until the combination is found by someone else
unlock check_mutex
if is_driver_found
   Remove seated threads from the line list.
```