

## Report

For PA3

Hasan Barış Aygen

At first, code checks if the number of team A and B fans are valid (they both have even and their sum can be divided by 4). Then it creates a thread list and a struct list. The struct is for keeping the team information. After that the main algorithm creates the threads by using a thread function and joins them one by one. The thread function works as, it checks for 4 conditions. Either A comes and there are 4 As or A comes and there are 2 As, 2 Bs or B comes and there are 4 Bs or B comes and there are 2 Bs, 2 As. If one of the 4 cases is valid, the last thread writes to console that it found a car and then waits for others to write. Finally, it declares itself as driver, prints the information to the console and decreases the number of fans for each team considering the number of fans in the car. If none of those cases are valid, it increases the number of fans by one. Those operations (incrementing and decrementing the values) are protected by mutex so that there will be only one thread in a certain time. Those numbers of fans are stored in structures of semaphores. Thanks to the barrier, in each step, the threads wait for each other and therefore the console output is in correct order.