RT0S Exercises in mutual exclusion II, semaphores and events

Exercise 1: Mutual exclusion

a) Use a mutex to secure mutual access to the screen.
b) Use a semaphore to secure mutual access to the screen.
c) Use a CriticalSection to secure mutual access to the screen
d) Try to use events to secure mutual access to the screen.

Exercise 2: Parking

Model cars as independent threads.
a)
Use a semaphore to control the access to a parking place with room for n cars.
A car outputs to the screen when it enters and leaves the parking place – preferably in a separate column on the screen. If you are lazy, then only make one car thread function using car_number as initialisation parameter.
b)
Now we have a parking cellar. A car is only allowed to enter if a free parking place is available. Furthermore, we only have one ramp (narrow road) for the up and down traffic in the parking cellar. Thus only one car is allowed at a time on this road inside. Model the cars as independent threads. A car outputs to the screen when it enters and leaves the parking place and when it enters and leaves the ramp inside the parking house – preferably in a separate column on the screen.
Use a combination of semaphores and mutexes to solve this problem.
c)
Now implement having two independent ramps for the parking cellar. If the first ramp is occupied, then a car must abandon the waiting after some time and try the other one instead.

Exercise 3 Calculator threads.

Make calculator threads. For the multiplication declare 3 global variables: mul_number_1 , mul_number_2 and mul_result. The multiply thread multiply mul_number_1 and mul_number_2 and assigns the result to the variable mul_result in an endless loop.
Do the same for addition, subtraction and division.
Now make a new thread performing a calculation using the calculator threads - for instance doing 4*3-2+8/2. This means it must assign 4 to mul_number_1 and 3 to mul_number_2, and pick up the result from mul_result. Do 4*3 and 8/2 “in parallel”.
Use windows events to handle the synchronization.