Exam 4

Question 1-

Mutual exclusion:
By examining the code, to enter in the CS, either the process 0 or 1 must read the value of the variable want, which is either 0 or 1.
Indeed, by examining the code, we have:
Read_0(want[1] = 1) → go-to line 1 or wait for want[1] to be 0.
Read_1(want[1] = 1) → go-to line 1 or wait for want[0] to be 0.
Then, since the value of want can be change only in the exit, we can be sure that one of the two processes must be blocked in the entry code since the second is in the CS.

Deadlock free:
The code doesn't provide deadlock freedom, as the next scenario shows:
Let the process 0 be interested then, by examining the code we have:
Write_0(want[0] = 0) → Read_0(priority = 0)   then here process 1 is interested and gets running time:
Write_1(want[1] = 0) → Read_1(want[0] = 0) → Write_1(want[1] = 1) here, process 0 gets running time.
Write_0(want[0] = 1) → Read_0(priority = 0) → Read_0(want[1] = 1) → stuck in the await loop.
Then, process 1 gets running time:
Read_1(priority = 0) → Read_1(want[0] = 1) → go-to line 1 → Write_1(want[1] = 0) → Read_1(want[0] = 1 and priority = 0) → stuck in the await loop.
Both processes stuck there is no deadlock freedom.

There is of course no starvation freedom since there is no deadlock freedom.