Assignment 5 OS.

A) False.

Like we saw during the lectures, in the claim 5.17 :
Peterson's algorithm is 1-bounded but is not 0-bounded.
That means that if the process 0 wait in the entry of the <CS> before the process 1, it's possible that the process 1 actually enter in the <CS> before the process 0.

We can show this with the next scenario :
1) The process 1 sets the turn_to_wait variable to 1.
2) The process 0 sets the turn_to_wait variable to 0 and enters wait area.
3) The process 1 enter in the <CS> and so bypass process 0.

B)
shared : boolean flag_for_p = False

Algorithm for q:
//Entry code
1. while (flag ==True);
2. Peterson_Enter_0(q, r)
3. // useless line
<CS>
//Exit code
1. flag = True
2. Peterson_Leave_0(q, r)

Algorithm for r:
//Entry code
1. while (flag == True);
2. Peterson_Enter_1(r, q)
3. // useless line
<CS>
//Exit code
1. flag = True
2. Peterson_Leave_1(r, q)

Algorithm for p:
//Entry code
1. while (flag == False);
<CS>
//Exit code
1. flag = False