

a) Não. Analisando as três casos possíveis: os dois processos chegando ao mesmo tempo no loop, o primeiro processo chegando antes do segundo e o segundo chegando antes do primeiro:

- Os dois chegam ao mesmo tempo: neste assim, o Processo 1 vai sair do loop e entrar na região crítica. Após sair da região crítica, o Processo 1 define $quer[1] := false$ e isso permite que o Processo 2 entre na região crítica.

- O Processo 1 chega primeiro no loop: similar ao item anterior.

- O Processo 2 chega primeiro no loop: o Processo 2 pode sair do loop pela condição do `if` (pois o $quer[1]$ ainda vai ter valor `false`). Então, ele entra na região crítica e marca o $quer[2] := false$ (nesse meio tempo podem entrar na região crítica ao mesmo tempo).

b) Sim. Supondo que o Processo 2 seja executado antes da atribuição $quer[1] := true$ no Processo 1, então, a verificação `quer[1]` vai retornar `True` (pois os valores de `quer` são inicialmente `false`). Ao mesmo tempo, e antes de executar a atribuição de $my := 2$, o Processo 1 pode ser executado e entrar na região crítica, logo após o Processo 2 sai do loop e também entra na região crítica. Portanto, o algoritmo proposto não satisfaz a propriedade de exclusão mútua.