

**1. (20) К** Проблем паркинга са заглављивањем. Постоје два паркинг места, при чему се возило паркирано на паркинг месту бр. 1 бива заглављено ако се попуни паркинг место бр. 2. Монитор треба да контролише ова два паркинг места по следећој логици: а) процеси аутомобили позивају процедуру *zelim\_da\_se\_parkiram* и паркирају се ако је неко од места слободно. У супротном чекају на ослобађање места за паркинг. Ако је паркинг празан, обавезно се паркирају на место 1. б) Када желе да напусте паркинг позивају процедуру *zelim\_da\_napustim\_parking* в) Ако су паркирани на месту бр. 1, морају да чекају ослобађање места бр. 2, да би отишли са паркинга. Написати монитор који решава овај проблем.

**2. (20) К** Решити проблем берберина који спава (*The Sleeping Barber Problem*). Берберница се састоји од чекаонице са  $N=5$  столица и берберске столице на којој се људи брију. Уколико нема муштерија, брица спава. Уколико муштерија уђе у берберницу и све столице су заузете, муштерија не чека, већ одмах излази. Уколико је берберин заузет, а има слободних столица, муштерија седе и чека. Уколико берберин спава, муштерија га буди. Користећи условне критичне регионе написати програм за берберина и муштерије.

**3. (20) И** Филтерска мрежа за сортирање са тернарним стаблом. Елемент мреже на своје улазе прима три сортирана низа која се завршавају са вредношћу EOS. Елемент треба да генерише излаз који је сортирани низ који се завршава са EOS. Приказати филтерску мрежу која треба да најбрже сортира 11 вредности.

**4. (20) И** Посматра се забавни парк са аутићима који могу да приме по једну особу (*Bumper Cars Problem*). Особа се шета по парку и може да одлучи да се провоза. Особа која жели да се вози стаје у ред и чека слободни ауто. Када се ослободи ауто и претходни возач напусти ауто прва особа из реда добија прилику да вози ауто и да се судара са другим аутима. Након неког времена особа завршава вожњу, враћа ауто на одговарајуће место, напушта ауто и одлази да се шета по парку. Особа може више пута да дође да се вози. Уколико постоји више слободних аута особа бира онај који је најдуже чекао на вожњу. Уколико тренутно нема особа које желе да се возе ауто стаје у ред и чека да дође особа. Користећи библиотеку *C-Linda* написати код за особу, ауто и за иницијализацију.

**1. (20) К** Проблем паркинга са заглављивањем. Постоје два паркинг места, при чему се возило паркирано на паркинг месту бр. 1 бива заглављено ако се попуни паркинг место бр. 2. Монитор треба да контролише ова два паркинг места по следећој логици: а) процеси аутомобили позивају процедуру *zelim\_da\_se\_parkiram* и паркирају се ако је неко од места слободно. У супротном чекају на ослобађање места за паркинг. Ако је паркинг празан, обавезно се паркирају на место 1. б) Када желе да напусте паркинг позивају процедуру *zelim\_da\_napustim\_parking* в) Ако су паркирани на месту бр. 1, морају да чекају ослобађање места бр. 2, да би отишли са паркинга. Написати монитор који решава овај проблем.

**2. (20) К** Решити проблем берберина који спава (*The Sleeping Barber Problem*). Берберница се састоји од чекаонице са  $N=5$  столица и берберске столице на којој се људи брију. Уколико нема муштерија, брица спава. Уколико муштерија уђе у берберницу и све столице су заузете, муштерија не чека, већ одмах излази. Уколико је берберин заузет, а има слободних столица, муштерија седе и чека. Уколико берберин спава, муштерија га буди. Користећи условне критичне регионе написати програм за берберина и муштерије.

**3. (20) И** Филтерска мрежа за сортирање са тернарним стаблом. Елемент мреже на своје улазе прима три сортирана низа која се завршавају са вредношћу EOS. Елемент треба да генерише излаз који је сортирани низ који се завршава са EOS. Приказати филтерску мрежу која треба да најбрже сортира 11 вредности.

**4. (20) И** Посматра се забавни парк са аутићима који могу да приме по једну особу (*Bumper Cars Problem*). Особа се шета по парку и може да одлучи да се провоза. Особа која жели да се вози стаје у ред и чека слободни ауто. Када се ослободи ауто и претходни возач напусти ауто прва особа из реда добија прилику да вози ауто и да се судара са другим аутима. Након неког времена особа завршава вожњу, враћа ауто на одговарајуће место, напушта ауто и одлази да се шета по парку. Особа може више пута да дође да се вози. Уколико постоји више слободних аута особа бира онај који је најдуже чекао на вожњу. Уколико тренутно нема особа које желе да се возе ауто стаје у ред и чека да дође особа. Користећи библиотеку *C-Linda* написати код за особу, ауто и за иницијализацију.