

**1. (20) К** Код *Tie breaker* алгоритма за  $n$  процеса се догодио следећи случај – приликом извршавања кода за улазак у критичну секцију, свих  $n$  процеса су ушли у стање 1 и ниједан још није ушао у стање 2. Одговорити на следећа питања и образложити одговор:

- а) Да ли процес који је први ушао у стање 1 први улази у стање 2?
- б) Да ли процес који је први ушао у стање  $n-2$  први улази у критичну секцију?
- в) Да ли процес који је последњи ушао у стање 1 може да буде трећи који улази у стање 2?
- г) Да ли процес који је последњи ушао у стање 1 може да буде први који улази у критичну секцију?

**2. (20) К** Посматра се агенција за изнајмљивање аутомобила. У возном парку постоје  $M$  стандардних возила и  $N$  лукс возила. Приликом доласка клијента који жели да изнајми ауто, он се изјашњава да ли жели да изнајми стандардно или лукс возило или му је свеједно. Клијент чека у реду све док му се возило не додели на коришћење. По завршетку коришћења, корисник долази још једном у агенцију и враћа ауто. Потребно је обезбедити да клијенти преузимају своје аутомобиле по редоследу доласка у агенцију. Клијент има могућност да изнајми више возила, али је сваки долазак везан искључиво за позајмицу једног аутомобила. Решити проблем користећи мониторе са *signal and continue* дисциплином.

**3. (20) И** У Линди реализовати лицитацију у којој постоји један процес *vođa licitacije* и  $n$  процеса *učesnika u licitaciji* и један процес *tick* који само ажурира време. Вођа лицитације треба да иницијализује торку са почетном вредношћу и да после датог временског интервала заврши лицитацију, успешно ако је излицитирана вредност већа од резервисане вредности, неуспешно иначе. Процеси учесници у лицитацији треба да се реализују тако да се не блокирају када се заврши дата лицитација, треба да знају која је излицитирана вредност и да ли су је они поставили.

**4. (20) И** Постоји  $N$  процеса *node(i:1..N)* који или чувају елементе листе или су слободни. Сваки од њих може да чува неку вредност (*value*) и показивач на следећи елемент у листи (*next*). Процес *head* чува показивач на први елемент у листи. Приликом брисања елемента из листе процес који га представља се пребацује на почетак листе слободних процеса на који указује процес *free*. Процес *start* иницира брисање елемента са задатом вредношћу из листе. Реализовати процедуре за све наведене процесе користећи CSP.

*Колоквијум + исцрпни трајаје 3 саћа.*

*Колоквијум трајаје 2 саћа. Исцрпни трајаје 2 саћа.*

**Напомена:** На вежбанци назначити који део се ради (**К**, **И** или **К+И**).

Дозвољено је користити готове структуре података (листе, редове, стек, хеш мапе, стабла, ...).

**1. (20) К** Код *Tie breaker* алгоритма за  $n$  процеса се догодио следећи случај – приликом извршавања кода за улазак у критичну секцију, свих  $n$  процеса су ушли у стање 1 и ниједан још није ушао у стање 2. Одговорити на следећа питања и образложити одговор:

- а) Да ли процес који је први ушао у стање 1 први улази у стање 2?
- б) Да ли процес који је први ушао у стање  $n-2$  први улази у критичну секцију?
- в) Да ли процес који је последњи ушао у стање 1 може да буде трећи који улази у стање 2?
- г) Да ли процес који је последњи ушао у стање 1 може да буде први који улази у критичну секцију?

**2. (20) К** Посматра се агенција за изнајмљивање аутомобила. У возном парку постоје  $M$  стандардних возила и  $N$  лукс возила. Приликом доласка клијента који жели да изнајми ауто, он се изјашњава да ли жели да изнајми стандардно или лукс возило или му је свеједно. Клијент чека у реду све док му се возило не додели на коришћење. По завршетку коришћења, корисник долази још једном у агенцију и враћа ауто. Потребно је обезбедити да клијенти преузимају своје аутомобиле по редоследу доласка у агенцију. Клијент има могућност да изнајми више возила, али је сваки долазак везан искључиво за позајмицу једног аутомобила. Решити проблем користећи мониторе са *signal and continue* дисциплином.

**3. (20) И** У Линди реализовати лицитацију у којој постоји један процес *vođa licitacije* и  $n$  процеса *učesnika u licitaciji* и један процес *tick* који само ажурира време. Вођа лицитације треба да иницијализује торку са почетном вредношћу и да после датог временског интервала заврши лицитацију, успешно ако је излицитирана вредност већа од резервисане вредности, неуспешно иначе. Процеси учесници у лицитацији треба да се реализују тако да се не блокирају када се заврши дата лицитација, треба да знају која је излицитирана вредност и да ли су је они поставили.

**4. (20) И** Постоји  $N$  процеса *node(i:1..N)* који или чувају елементе листе или су слободни. Сваки од њих може да чува неку вредност (*value*) и показивач на следећи елемент у листи (*next*). Процес *head* чува показивач на први елемент у листи. Приликом брисања елемента из листе процес који га представља се пребацује на почетак листе слободних процеса на који указује процес *free*. Процес *start* иницира брисање елемента са задатом вредношћу из листе. Реализовати процедуре за све наведене процесе користећи CSP.

*Колоквијум + исцрпни трајаје 3 саћа.*

*Колоквијум трајаје 2 саћа. Исцрпни трајаје 2 саћа.*

**Напомена:** На вежбанци назначити који део се ради (**К**, **И** или **К+И**).

Дозвољено је користити готове структуре података (листе, редове, стек, хеш мапе, стабла, ...).