

Strukture podataka i algoritmi 1
Oktober 2, 2009

1. Studentski dom je od kompanije IMI-Soft naručio softver za vođenje evidencije o sobama u studentskom domu, tako da on zadovolji sledeće zahteve:

- a. U programu je moguće voditi evidenciju o studentima i sobama.
Svaki **student** je definisan matičnim brojem, imenom i pokazivačem na sobu u koju je smešten.
Svaka **soba** je definisana rednim brojem i brojem kreveta.
(3 poena)

Podaci o rasporedu studenata po sobama su zapisani u datoteci čiji je format:

```
n          //broj soba
rbs1 brk1
rbs2 brk2
...
rbsn brkn
m          // broj studenata
mat1
ime1
rbs1
mat2
ime2
rbs2
...
matm
imem
rbsm
```

Napisati funkciju *UčitajSobe* koja iz zadate datoteke učitava podatke o sobama.
(2 poena)

Napisati funkciju *PronađiSobu* koja vraća pokazivač na sobu sa zadatim rednim brojem.
(3 poena)

Napisati funkciju *UčitajStudente* koja iz zadate datoteke učitava podatke o studentima i svakog studenta povezuje sa odgovarajućom sobom.
(4 poena)

- b. Napisati funkciju *ObrišiStudenta* koja iz evidencije briše studenta sa zadatim matičnim brojem.
(5 poena)
- c. Napisati funkciju *ProveriDupliranje* koja proverava da li je neki student unet više puta u evidenciju i ako jeste, briše suvišna pojavljivanja tog studenta.
(5 poena)
- d. Napisati funkciju *ObrišiSobu* koja iz evidencije uklanja sobu sa zadatim rednim brojem, a sve studente koji su u njoj boravili premešta u poseban spisak neraspoređenih studenata.
(6 poena)
- e. Napisati program u kome se zadaje naziv datoteke u kojoj se nalaze podaci o sobama i studentima, a zatim proveriti da li postoji dupliranje studenata i ukoliko postoji obrisati suvišna pojavljivanja. Nakon toga ponuditi korisniku da unese redni broj sobe koju treba obrisati iz evidencije, obrisati zadatu sobu, a studente koji su u njoj boravili prebaciti u spisak neraspoređenih studenata.
(2 poena)