

Zadatak 1 Ime datoteke dato je kao argument komandne linije. U datoteci se nalaze otvorene i zatvorene zagrade i još nekakav tekst. Proveriti da li su zagrade pravilno uparene. Npr. `ab(cd) ..` odgovor je `jesu`, a `..)ba()` odgovor je `nisu`.

Zadatak 2 Napraviti strukturu `STUDENT` koja sadrži:

- `ime` (u polju se čuva ime i prezime studenta, napr. "Marko Markovic", maksimalna dužina polja je 100 karaktera),
- `oc` (sadrži 10 ocena studenta)
- `pr_oc` (prosečna ocena)

U datoteci `ocene.txt` nalazi se broj `n`, a potom podaci za `n` studenata. Za svakog studenta unosi se ime i prezime razdvojeno razmakom (uputstvo: može se koristiti `gets`), a potom u sledećem redu deset ocena. Pronaći studenta koji ima najveći prosek i ispisati sve njegove podatke. Uraditi koriscenjem dinamičke alokacije.

Zadatak 3 Napisati program koji prepisuje datoteku čije se ime navodi kao prvi argument komadne linije u datoteku čiji se argument navodi kao drugi argument komadne linije programa. Ukoliko je navedena opcija `-u` program prilikom prepisivanja treba da zamenjuje sva mala slova velikim, a ukoliko je navedena opcija `-l` sva velika slova se zamenjuju malim. Ako nisu navedene opcije izvršiti samo prepisivanje iz jedne u drugu datoteku.

Zadatak 4 a) Napisati C funkciju `int unesiSkup(char *s, FILE* f)` kojom se unosi skup elemenata iz datoteke `F`. Skup se predstavlja kao niz karaktera, pri čemu su dozvoljeni elementi skupa mala i velika slova abecede, kao i cifre. Unos se prekida kada se nađe na znak za novi red ili nedozvoljeni karakter za skup (ne zna se koliko skup može imati elemenata). Funkcija vraća broj elemenata skupa koji su uspesno učitani.

b) Napisati funkciju `void prebroj(char *s, int *br_slova, int *br_cifara)` kojom se određuje broj slovnih elemenata skupa (velikih ili malih slova) kao i broj cifara u skupu.

c) Napisati glavni program gde se unose podaci o skupu elemenata. Ime datoteke se zadaje kao argument komandne linije. Na standardni izlaz ispisati informacije o broju slova i cifara (koristiti funkcije pod a) i b)).

Zadatak 5 Napisati funkciju `int podmatrica(int** a, int n, int** b, int m)`, koja računa da li je matrica `b` podmatrica matrice `a`. Matrice su kvadratne i `n` je dimenzija matrice `a`, a `m` je dimenzija matrice `b`. Matrica `B` je podmatrica matrice `A`, ako izostavljanjem nekih vrsta i kolona matrice `A` možemo dobiti matricu `B`.

Napr. Neka je $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ i $B = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$. Tada je `B` podmatrica matrice

A. Sa standardnog ulaza unose se matrice. Prvo se unose dimenzije matrica, a potom i same matrice. Maksimalna dimenzija matrice nije unapred poznata.

Zadatak 6 Uobičajena praksa na UNIX sistemima je da se argumenti komandne linije dele na opcije i argumente u užem smislu. Opcije počinju znakom - nakon čega obično sledi jedan ili više karaktera koji označavaju koja je opcija u pitanju. Ovim se najčešće upravlja funkcionisanjem programa i neke mogućnosti se uključuju ili isključuju. Argumenti načšće predstavljaju opisne informacije poput na primer imena datoteka. Napisati program koji ispisuje sve opcije koje su navedene u komandnoj liniji. Na primer, ako se program pokrene sa

```
./echoopts -abc input.txt -d -Fg output  
program treba da ispiše
```

Prisutne su opcije:
a b c d F g

Zadatak 7 Dimenzije matrice i ime datoteke su argumenti komadne linije. Napisati program koji učitava matricu sa standardnog ulaza, a potom ispisuje spiralno elemente matrice u datoteku. Napr. za matricu

```
1  2  3  4  5  
6  7  8  9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25
```

Treba ispisati: 1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6 7 8 9 14 19 18 17 12
13