

**Uvod u programiranje**

Ispit – februarski rok – grupa 1

1. **(13p)** Napisati program koji učitava matricu dimenzija NxN i na osnovu učitane matrice formira novu matricu na sledeći način. Svaki element u novoj matrici dobija vrednost broja različitih prostih brojeva koji okružuju element sa njegovom pozicijom u originalnoj matrici. Elementi koji se nalaze na ivici matrice su okruženi na sledeći način:

Uvesti „cikluse“, tako da je poslednja kolona „levi“ sused prve kolone, a poslednji red „gornji“ sused prvog reda (efektivno torus).

Na primer, ako bismo posmatrali element na poziciji [0,0] u matrici dimenzija 4x4, njegovi susedi su na pozicijama: [3,3], [3,0], [3,1], [0,1], [1,1], [1,0], [1,3], [0, 3].

Primeri:

Ulez: 4

6	3	8	1
5	7	2	3
9	6	4	7
2	1	5	9

Izlaz:

5	4	5	3
3	3	3	4
5	4	5	3
3	3	3	4

Ulez: 5

7	2	1	5	8
4	3	6	2	1
8	2	5	7	2
3	6	5	9	4
9	7	1	3	2

Izlaz:

4	3	5	3	5
4	4	5	4	3
3	2	4	3	4
2	5	5	5	3
3	5	5	3	3

2. **(12p)** Napisati program koji učitava string koji sadrži reči razdvojene spejsom i unutar njega vrši sledeće izmene, bez upotrebe pomoćnih nizova (stringova). Svaku reč čijih pola ili više slova su samoglasnici uklanja iz stringa, a sve ostale reči ispisuje velikim slovima.

Primeri:

Ulez: Ne postoji put ka miru , mir je put

Izlaz: POSTOJI PUT MIR PUT

Ulez: Svaki kraj je neki novi pocetak

Izlaz: SVAKI KRAJ POCETAK

3. **(25p)** Deklarisana je struktura Proizvod na sledeći način

```
typedef struct Proizvod{
    char naziv[25]; int cena; struct Proizvod *sledeci;
} Proizvod;
```

U tekstualnoj datoteci "proizvodi.txt" nalaze se informacije o proizvodima u marketu i njihovim cenama.

Svaki red datoteke sadrži naziv proizvoda i njegovu cenu, razdvojene povlakom (-).

U fajlu zadatka (koji se kompajlira ali ne radi ništa) je dat šablon za funkcije koje je potrebno da se implementiraju (ne menjati im potpis!), i glavni program koji ne treba menjati. Implementirati sledeće funkcije (dozvoljeno je dodavati i pomoćne funkcije):

a) **(10p)** `Proizvod *ucitajFajl()` - učitava datoteku u listu i vraća pokazivač na početak liste.

Svi proizvodi iz datoteke prilikom učitavanja je potrebno da budu sortirani alfabetски rastuće.

Dakle, na početku liste nalazi se proizvod čiji je naziv "najmanji" po azbučnom/redosledu karaktera, a zatim slede ostali. Ako je lista prazna, njen početak je NULL.

- b) **(2p) void ispisiListu(Proizvod \*lista)** - ispisuje sve proizvode iz liste sortirano po nazivu opadajuće (tj. od kraja ka početku u odnosu na to kako su proizvodi učitani u listu). Ispis na ekranu može biti u formatu kao iz datoteke. Ispis od početka do kraja umesto od kraja do početka vredi 50% poena.
- c) **(4p) double prosecnaCena(Proizvod \*lista)** - vraća prosečnu vrednost cena svih proizvoda iz liste. Ako je lista prazna, neka funkcija vrati nulu.
- d) **(7p) void izbacSiSkuplje(Proizvod \*\*lista, int cena)** - izbacuje iz liste (briše) sve proizvode koji su skuplji od zadate cene. Voditi računa o pravilnom oslobođanju memorije. Ako je posle ove operacije lista prazna, pokazivač na početak treba da postane NULL.
- e) **(2p) void brisanje(Proizvod \*\*lista)** - briše sve elemente liste (oslobađa svu zauzetu memoriju), postavlja početak liste na NULL.

Spisak dozvoljenih bibliotečkih funkcija (u nastavku) važi generalno. Dozvoljeno je implementirati svoju verziju bibliotečkih funkcija koje nisu dozvoljene.

- **stdio.h:** printf, scanf, fscanf, fgets, puts, fputs
- **string.h:** strlen, strcpy, strcat, strstr, strchr, strcmp, strtok
- **math.h:** sqrt, abs, pow, ceil, floor, round
- **sdlib.h:** malloc, calloc, realloc, free, atoi, atof

#### **Primer ispisa za treći zadatak:**

##### Početak ispisa:

Boranija, 50

Brasno, 70

Breskva, 60

Cokolada, 150

....

##### Ispis prosečne cene:

Prosecna cena: 94.5

....

##### Ispis posle izbacivanja proizvoda skupljih od 100:

Boranija, 50

Brasno, 70

Breskva, 60

Jaja, 90

....

##### Posle brisanja:

(nil)