

**Uvod u programiranje**

## Ispit – julski rok - grupa 1

1. **(13p)** Napisati program koji učitava broj  $N$  i matricu  $A$  dimenzija  $N \times N$  i od njenih elemenata formira novu matricu  $B$  dimenzija  $N \times N$  na sledeći način: polje  $B[i][j]$  ima vrednost najdeljivijeg od svih elemenata koji se nalaze u vrsti  $i$  i koloni  $j$  osim elementa  $A[i][j]$ . Najdeljiviji je broj sa najvećim brojem delilaca (uključujući 1 i taj broj). Sve vrednosti upisati u matricu  $B$  i ispisati je u glavnom programu. Ako za neko  $(i,j)$  ima više najdeljivijih elemenata, uzeti najmanji. Korišćenje pomoćnih nizova nije dozvoljeno. Primer:

Ulaz: 5

10 17 12 22 28

27 16 37 50 14

36 30 40 44 60

20 11 13 21 16

12 24 56 95 46

Izlaz:

36 24 40 12 60

36 24 40 44 60

60 60 60 60 36

36 24 40 20 60

36 30 24 24 60

Objašnjenje je na drugoj strani.

2. **(12p)** Napisati program koji učitava jedan red teksta (proizvoljni simboli) i prepravlja ga na sledeći način:
- za svako pojavljivanje sekvence  $X+$  (gde je  $X$  bilo koji karakter osim '+', '?' i '!'), umesto te sekvence upisuje  $XXX$  (tj. tri puta simbol koji je bio pre znaka '+')
  - za svako pojavljivanje sekvence  $X?$  (gde je  $X$  bilo koji karakter osim '+', '?' i '!'), umesto te sekvence upisuje  $Y$ , gde je  $Y$  odgovarajuće malo slovo ako je  $X$  veliko slovo, veliko slovo ako je  $X$  malo, komplement od  $X$  do zbira 10 ako je  $X$  cifra, ili samo  $X$  ako  $X$  nije ni slovo ni cifra
  - za svako pojavljivanje sekvence  $!X$  (gde je  $X$  bilo koji karakter osim '+', '?' i '!'), uklanja tu sekvencu i naredna dva karaktera iz stringa (ili manje, ako do kraja stringa nema dovoljno karaktera za brisanje); ako je  $!$  poslednji simbol u stringu, ne uklanja se.

Program ispisuje prepravljeni string. Sve operacije raditi direktno nad ulaznim stringom (nije dozvoljeno koristiti dodatne stringove / nizove). String mora da bude potpuno učitán pre obrade (ne obrađivati string prilikom učitavanja). Garantovano u ulaznom stringu neće biti uzastopnih specijalnih karaktera ('+', '?' i '!').

Primer:

Ulaz: **d?onacijama solidarnosti! je prikupljeno je 1?0000+ dinara -? za N?arodnu kuhinju.+**Izlaz: **Donacijama solidarnosti prikupljeno je 9000000 dinara - za narodnu kuhinju...**

3. **(25p)** Deklarisane je struktura `Element` na sledeći način:

```
typedef struct Element {
    char naziv[25]; int broj; struct Element *sledeci;
} Element;
```

U tekstualnoj datoteci "ulaz.txt" nalazi se lista gradova sa brojem stanovnika. Svaki red datoteke sadrži ime grada i broj stanovnika razdvojene zapekom.

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

a) (10) **Element \* ucitajFajl()** - učitava datoteku u listu i vraća pokazivač na početak liste. Sve gradove iz datoteke prilikom učitavanja sortirati opadajuće po broju stanovnika (na početku liste je grad sa najviše stanovnika, svaki sledeći ima manje). Voditi računa o praznom redu na kraju datoteke. Ako je lista prazna, njen početak je NULL.

b) (2) **void ispisListu(Element \*lista)** - ispisuje sve gradove iz liste sortirano po broju stanovnika rastuće (tj. od kraja ka početku). Ispis na ekranu može biti u formatu kao iz datoteke. Ispis od početka do kraja umesto od kraja do početka vredi 50% poena.

c) (5) **int brojGradova(Element \*lista, char \*str)** - vraća broj gradova u listi čiji naziv počinje na string str (osetljivo na velika i mala slova).

d) (6) **void izbaciMale(Element \*\*lista, int broj)** - izbacuje iz liste sve gradove koji nemaju više od zadatog broja stanovnika. Voditi računa da su elementi pravilno obrisani (da je oslobođena memorija). Ako je obrisana cela lista, posle ove funkcije trebalo bi da postane NULL.

e) (2) **void brisanje(Element \*\*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 nabrojane.

- **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
- **stdlib.h:** malloc, calloc, realloc, free, atoi, atof

#### Pojašnjenje primera za prvi zadatak:

Broj delilaca svakog elementa matrice A dat je u sledećoj matrici:

4 2 6 4 6

4 5 2 6 4

9 8 8 6 12

6 2 2 4 5

6 8 8 4 4

Tako recimo za polje B[0][1] razmatra se broj delilaca za elemente iz vrste 0 i kolone 1 (podebljani brojevi). Najveći broj delilaca je 8. Ima dva elementa sa tim brojem delilaca, to su A[2][1] = 30 i A[4][1] = 24, pa se u B[0][1] upisuje manji od njih (24).

#### Pojašnjenje primera za drugi zadatak:

Ulaz: d?onacijama solidarnosti! je prikupljeno je 1?0000+ dinara -? za N?arodnu kuhinju.+

Izlaz: Donacijama solidarnosti prikupljeno je 9000000 dinara - za narodnu kuhinju...

Za "d?" važi da pošto je d malo slovo, treba da postane veliko, tj. umesto te sekvence upisuje se D.

Za "!" je " " važi da se ta sekvenca uklanja (tj. uzvičnik, razmak i još dva slova posle). Za "1?" pošto je 1 cifra, treba je dopuniti do 10, tj. umesto te sekvence upisuje se 10-1=9. Za "0+" potrebno je zameniti tu sekvencu sa tri nule, tj. "000". Za "-?", pošto "-" nije ni slovo ni cifra, ostaje kako jeste, ali se upitnik uklanja. Za "N?" pošto je N veliko slovo, menja se malim. Za ".+" tačka se utrostručuje.

**Primer ispisa za treći zadatak:**

Početak ispisa:

Crna Trava, 1063  
Trgovište, 4316  
Gadžin Han, 5850  
Bosilegrad, 6065  
Medveđa, 6360

....

Kraj prvog ispisa i prelaz na operacije:

Grad Niš, 249501  
Novi Sad, 368967  
Grad Beograd, 1681405

Broj gradova: 5  
Broj gradova na No: 6

Posle izbacivanja  
Bor, 40845  
Bujanovac, 41068  
Arandjelovac, 41297

....

Kraj drugog ispisa i brisanje:

Subotica, 123952  
Kragujevac, 171186  
Grad Niš, 249501  
Novi Sad, 368967  
Grad Beograd, 1681405

Posle brisanja  
(nil)