



CS130/CS323: C/C++ programski jezik

DOMAĆI ZADATAK 5.

Prilikom slanja domaćeg zadatka neophodno je da ispunite sledeće:

- Subject mail-a mora biti CS323-DZbr (u slučaju kada šaljete domaće za ovu nedelju to je CS323-DZ05)
- U prilogu maila treba da se nalazi arhiviran (zip, rar, ...) projekat koji se ocenjuje, imenovan na sledeći način: CS323-DZbr-ImePrezimeBrojIndeksa.
Na primer, CS323-DZ05-VeljkoGrkovic1234
- Potrebno je uraditi i printscreen koda programa sa rezultatom koji program daje
- Telo mail-a treba da ima pozdravnu poruku

Molimo sve studente da se pridržavaju navedenog, inače zadaci neće biti pregledani i ocenjeni.

Studenti iz Niša zadatak na pregled šalju na mail:

tamara.vukadinovic@metropolitan.ac.rs

Studenti iz Beograda i internet studenti zadatak na pregled šalju na mail:

lazar.mrkela@metropolitan.ac.rs

Svaki student radi po dva zadatka sa spiska. Ostali zadaci mogu da posluže za vežbanje i pripremu ispita, ali ih ne šaljete profesoru ili asistentima na pregledavanje.

Student bira zadatke po sledećim formulama:

Zadatak 1: Broj indeksa % 20 + 1 (Npr. $2378 \% 20 + 1 = 19$ – Student radi 19. zadatak).

Zadatak 2: Broj indeksa % 20 + 21 (Npr. $2378 \% 20 + 21 = 39$ – Student radi 39. zadatak).

Grupa zadataka 1:

1. Napisati program koji učitava n MP3 **pesama** (naziv, izvođač, godina izvođenja). Prikazati koliko od tih pesama je izdato u 2017. godini.
2. Napisati program koji učitava n **osoba** (ime, adresu, datum rođenja formata dd.mm.yy). Posle unosa podataka, program prikazuje osobu koja je rođena u znaku raka, a koja je najstarija (21.06 - 21.07). Ukoliko se dogodi da ima više osoba koje su rođene istog datuma (a najstarije) oštampati prvu na koju se naiđe.
3. Data je struktura

```
struct complex {  
    double re;  
    double im;  
}
```

Napisati funkciju koja za dati kompleksan broj nalazi njemu konjugovano kompleksni broj. U glavnom programu testirati rad funkcije tako sto se unese niz od N brojeva, a njima konjugovano kompleksni brojevi se stampaju na ekran.
Konjugovan kompleksan broj je broj kome je promenjen znak imaginarnog dela
4. Napisati program kojim se korišćenjem funkcije:
void citaj(struct licnost *osoba) učitavaju podaci za dve osobe, zatim adresa strukturne promenljive u kojoj su podaci starije osobe predaje pokazivačkoj promenljivoj s i ispisuju elementi strukture na koju ova promenljiva pokazuje.
Struktura osoba treba da sadrži podatke:char ime[30], char adresa[50], int starost.
5. Napisati program koji učitava niz od n elemenata strukture **ličnost**, koja sadrži sledeća polja: ime, adresa, dan rođenja, mesec rođenja i godina rođenja. Unete elemente prikazati na ekranu. Prikazati osobu koja je rođena u mesecu maju, a najmlađa od svih ostalih rođenih u maju.
6. Napisati program koji učitava niz od n elemenata strukture **aranžman** koja sadrži sledeće podatke: ime agencije, ime hotela, mesto, broj noćenja, cena aranžmana. Ispisati podatke o aranžmanu čije je mesto letovanja "Lefkada", a cena aranžmana najveća.
7. Napisati funkciju koja oduzima dva kompleksna broja i rezultat vraća preko liste argumenata (kao pokazivač). **Kompleksni broj** predstaviti korišćenjem strukture. Testirati rad funkcije u glavnom programu.
8. Data je struktura

```
struct Student {  
    char ime[10];  
    char prezime[10];
```

```
char smer;  
double prosek;  
}
```

a) Napisati funkciju koja učitava sa standardnog ulaza podatke o studentu.

b) Napisati funkciju koja ispisuje podatke o studentu na standardni izlaz.

U glavnom programu učitati niz od 5 studenata i ispisati podatke o studentu sa najvećim prosekom.

9. Data je struktura

```
struct complex{  
    double re;  
    double im;  
}
```

Napisati funkciju koja računa moduo kompleksnog broja. U glavnom programu testirati rad funkcije tako što se unese niz od N brojeva, a njihov moduo se štampa na ekran.

10. Kreirati strukturu **Datum**, a zatim je koristiti u strukturi **Ličnost** koja treba da sadrži podatke: char ime[30], char adresa[50], Datum datum. Kreirati niz od n ličnosti. Prikazati osobu koja je rođena 1995. godine, a najstarija (od svih ostalih rođenih 1995. godine).

11. Data je struktura

```
struct complex {  
    double re;  
    double im;  
}
```

Napisati funkciju koja izračunava zbir i razliku dva kompleksna broja. U glavnom programu testirati rad funkcije.

12. Napisati program kojim se učitava niz struktura deklarisan sa:

```
Struct licnost osoba[MAXOS], *pok;
```

i ispisuje na dva načina. Prvo korišćenjem elemenata niza, a zatim korišćenjem pokazivača koji se inicijalizuje adresom niza struktura. Struktura treba da sadrži podatke: char ime[30], char adresa[50], int starost.

13. Definirati 2D **Vektor** u prostoru korišćenjem strukture. Napisati funkcije koje računaju zbir, razliku, skalarni proizvod i vektorski proizvod dva vektora.

14. Deklarirati strukturu **Automobil** koja ima podatke: model, godina proizvodnje, cena, pređena kilometraža. Učitati podatke za n automobila u glavnom programu. Napisati funkciju koja prikazuje podatak o automobilu čija je godina proizvodnje 2000, a cena najmanja.

15. Definirati strukturu **student** koja sadrži sledeće podatke: ime, smer, prosek. Kreirati niz od n studenata. U glavnom programu uneti realan broj p i ispisati podatke o studentima čiji je prosek veći ili jednak unetom broju p.
16. Definirati strukturu **aranžman** koja sadrži sledeće podatke: ime agencije, ime hotela, mesto, broj noćenja, cena aranžmana. Kreirati niz od n aranžmana. Ispisati podatke o aranžmanu čiji je broj noćenja 10, a cena aranžmana najmanja.
17. Definirati strukturu **roditelj** koja sadrži sledeće podatke: ime roditelja, ime deteta. Program sadrži funkciju Broj dece koja za ime roditelja koje joj je prosleđeno pri pozivu vraća koliko taj roditelj ima dece. U glavnom programu učitati podatke o roditeljima i ispisati ime roditelja sa najvećim brojem dece (ako ih ima više ispisati sva imena).
18. Napisati program kojim se učitavaju podaci za 5 **osoba** i ispisuju podaci o najstarijoj, kao i podaci koja ima najmanje prvo slovo u imenu (najmanje je 'a' ili prvo najbliže slovu 'a'). Struktura treba da sadrži podatke: char ime[30], char adresa[50], int starost;
19. Definirati strukturu **ugovor** koja sadrži sledeće podatke: ime firme, vrednost ugovora. Program sadrži funkciju Ugovori koja dobija naziv firme, a vraća ukupnu vrednost svih ugovora potpisanih sa tom firmom. U glavnom programu učitati podatke i za svaku firmu ispisati ukupnu vrednost ugovora.
20. Napisati program kojim se učitavaju N **fudbalskih ekipa**. Svaka fudbalska ekipa ima podatke: naziv, broj bodova, gol razlika. Odrediti ekipu sa najboljim skorom i njene podatke oštampati na ekran. Ukoliko dve ekipe imaju isti broj bodova gleda se bolja gol razlika.

Grupa zadataka 2:

21. Napisati funkciju u kojoj se formira novi string koji se dobija tako što se iz originalnog stringa sve višestruke praznine menjaju jednom prazninom. Novi string kreirati kao dinamički niz u okviru funkcije, a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
22. Napisati funkciju kojom se izbacuje svako pojavljivanje karaktera c iz učitano stringa. Novodobijeni string formirati kao dinamički niz u okviru funkcije, a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
23. Formirati niz A[] od n sličajno odabranih elemenata iz intervala od 0-100. Napisati funkciju koja takav niz deli na dva niza: niz B[] sadrži negativne a niz C[] pozitivne elemente. Kreirati nizove B i C korišćenjem dinamičkog alociranja memorije, a kao argumente funkcije vratiti ova dva niza kao i njihove dimenzije. Testirati rad funkcije u glavnom programu.

24. Napisati program koji za unetu dimenziju i elemente niza x, pravi novi niz y tako što se u njega iz x prepisuju elementi sledećim redom: prvi, poslednji, drugi, pretposlednji,...
Ispisati niz y. Raditi isključivo sa dinamički alociranim nizovima.
25. Napisati funkciju koja od stringa s formira novi u kome se redom izbacuje svaki znak *, a svaki znak koji se od njega razlikuje duplira. Novi string kreirati kao dinamički niz u okviru funkcije, a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
26. Formirati dva niza A[] i B[] od m i n elemenata. Napisati funkciju u kojoj se formira treći niz C[] od dva uneta niza tako što svi negativni elementi nizova A i B postaju elementi trećeg niza. Memorija za niz C se alokira u okviru funkcije. Pokazivač na dinamički alocirani niz vratiti kao rezultat funkcije. Kao argument funkcije vratiti korišćenjem pokazivača dimenziju novonastalog niza. Testirati funkciju u glavnom programu.
27. Napisati funkciju koja će da služi za "sažimanje" niza, tako da se svaki podniz nula zameni jednom nulom. Funkcija kao rezultat vraća dinamički niz koji se dobija sažimanjem početnog niza.
28. Napisati funkciju koja nalazi sve proste brojeve manje od broja N. Funkcija treba glavnom programu kroz listu argumenata da vrati dinamički niz koji sadrži te proste brojeve i broj prostih. Napisati i glavni program koji učitava broj N i štampa niz prostih brojeva kao i koliko ih ima.
29. Napisati funkciju kojom se na osnovu N različitih rezultata trkača na 100m datih nizom a[i], i=1,n, (indeks odgovara startnom broju) formira niz b[i], b=1,n gde je vrednost b[i] indeks(startni broj) takmičara koji se plasirao na j-to mesto. Raditi isključivo sa dinamički alociranim nizovima.
30. Napisati program kojim se učitava ceo broj n (n<20), dva celobrojna niza a i b dužine n i potom formira niz c naizmeničnim mešanjem elemenata nizova a i b. Raditi isključivo sa dinamički alociranim nizovima.
Na primer:
ako korisnik unese n = 5 i
a: 1 2 5 6 9
b: 6 3 1 8 0
rezultujući niz je: 1 6 2 3 5 1 6 8 9 0
31. Napisati funkciju void zameni(char s[]) koja u stringu s maksimalne dužine 20 karaktera zamenjuje svako pojavljivanje slova a slovom b.
Na primer: string dan se zamenjuje stringom dbn.
U glavnom programu učitati string sa ulaza i testirati funkciju. Raditi isključivo sa dinamički alociranim nizovima.

32. Napisati funkciju `char * zameni3(char s[])` koja zamenjuje svaki karakter stringa `s` sa 3 naredna karaktera po abecedi i tako dobijen string korišćenjem dinamički alociranog niza vraća u glavni program. Na primer: string ANA treba modifikovati u BCDOPQBCD. Napisati potom program koji sa ulaza učitava 5 stringa maksimalne duzine 20 karaktera i ispisuje odgovarajuće modifikacije.
33. Napisati funkciju koja izdvaja sve parne elemente unetog niza u drugi dinamički alocirani niz. Pokazivač na dinamički alocirani niz vratiti kao rezultat funkcije. Kao argument funkcije vratiti korišćenjem pokazivača i dimenziju novonastalog niza. Testirati funkciju u glavnom programu.
34. Napisati funkciju u kojoj se formira string `s3` umetanjem stringa `s2` u string `s1` počev od pozicije `p`. Novodobijeni string `s3` formirati kao dinamički niz u okviru funkcije, a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
35. Napisati funkciju koja izdvaja sve pozitivne elemente unetog niza u drugi dinamički alocirani niz. Pokazivač na dinamički alocirani niz vratiti kao rezultat funkcije. Kao argument funkcije vratiti korišćenjem pokazivača i dimenziju novonastalog niza. Testirati funkciju u glavnom programu.
36. Napisati funkciju `char *slova(char *s)` koja vraća string u kojem su izbačeni svi karakteri iz `s` osim slova. Novi string kreirati kao dinamički niz u okviru funkcije a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
37. Napisati funkciju kojom se izbacuju sve praznine u učitanoj stringu. Novodobijeni string formirati kao dinamički niz u okviru funkcije, a zatim ga vratiti u glavni program. Testirati rad funkcije u glavnom programu.
38. Formirati niz od `N` elemenata, od kojih je prvi jednak `M`, a svaki sledeći član je jednak zbiru kubova prethodnog elementa niza. Raditi isključivo sa dinamički alociranim nizovima.
39. Napisati funkciju koja izdvaja sve neparne elemente unetog niza u drugi dinamički alocirani niz. Pokazivač na dinamički alocirani niz vratiti kao rezultat funkcije. Kao argument funkcije vratiti korišćenjem pokazivača i dimenziju novonastalog niza. Testirati funkciju u glavnom programu.
40. Formirati dva niza `A[]` i `B[]` od `m` i `n` elemenata. Napisati funkciju u kojoj se formira treći niz `C[]` od dva uneta niza tako što svi parni elementi nizova `A` i `B` postaju elementi trećeg niza. Memorija za niz `C` se alocira u okviru funkcije. Pokazivač na dinamički alocirani niz vratiti kao rezultat funkcije. Kao argument funkcije vratiti korišćenjem pokazivača dimenziju novonastalog niza. Testirati funkciju u glavnom programu.