



CS323: C/C++ programski jezik

DOMAĆI ZADATAK 4.

Prilikom slanja domaćeg zadatka neophodno je da ispunite sledeće:

- Subject mail-a mora biti CS323-DZbr (u slučaju kada šaljete domaći za ovu nedelju to je CS323-DZ04)
- U prilogu maila treba da se nalazi arhiviran (zip, rar, ...) projekat koji se ocenjuje, imenovan na sledeći način: CS323-DZbr-ImePrezimeBrojIndeksa.
Na primer, CS323-DZ04-VeljkoGrkovic1234
- Poželjno je uraditi i print-screen koda pre pokretanja programa i dodati u arhivu sa zadacima
- Telo mail-a treba da ima pozdravnu poruku

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

Studenti iz Beograda rešenja domaćih zadataka šalju mail-om asistentima Aleksandri Arsić ili Lazaru Mrkeli, u zavisnosti kojoj grupi za vežbe pripadaju.

Studenti iz Niša rešenja domaćih zadataka šalju mail-om asistentu Veljku Grkoviću.

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).

1. Napisati program koji tri broja x, y, z uređuje u ne-rastući poredak $x \geq y \geq z$. U program uključiti funkciju ***void uredi()*** čiji su argumenti pokazivači na adrese promenljivih a, b i kojom se razmenjuju vrednosti dve promenljive.
2. Napisati funkciju ***void minmax()*** kojom se izračunavaju minimum i maksimum tri zadata broja a, b, c. Funkciju ***minmax()*** realizovati korišćenjem pokazivača kao argumenata (npr. *min i *max).
3. Napisati funkciju koja pronalazi najveći zajednički delilac dva cela broja. Funkcija treba da vrati vrednosti najvećeg zajedničkog delioca kroz listu argumenata:
void NZD(int a, int b, int *nzd);
4. Napisati C funkciju ***void broj_ncifara(int n, int *brojCifara);*** kojom se izračunava broj neparnih cifara (1, 3, 5, 7, 9) u zapisu broja n. U main funkciji učitalavati brojeve sa ulaza sve dok korisnik ne unese negativan broj i na izlazu ispisivati broj pojavljivanja neparnih cifara.
5. Napisati C funkciju ***void broj_pcifara(int n, int *brojCifara);*** kojom se izračunava broj parnih cifara (0, 2, 4, 6, 8) u zapisu broja n. U main funkciji učitalavati brojeve sa ulaza sve dok korisnik ne unese negativan broj i na izlazu ispisivati broj pojavljivanja parnih cifara.
6. Napisati funkciju ***void broj_dana(int m, int *broj);*** kojom se izračunava broj dana u mesecu m. U slučaju meseca februara treba pitati korisnika o kojoj godini se radi i nakon analize da li je godina prestupna formirati korektan rezultat. Funkciju realizovati korišćenjem ***switch*** naredbe. Zatim u main funkciji učitalati redni broj meseca (mora biti iz intervala od 1 do 12) i ispisati njegov broj dana.
7. Napisati funkciju ***void zbir_delilaca (int n, int *);*** kojom se izračunava zbir delilaca broja n. Zatim u main funkciji učitalati broj k i na ekranu ispisati zbir delilaca svakog od brojeva iz intervala od 1 do k.
8. Napisati program koji formira niz od N elemenata i određuje minimalni element niza kao i njegovu poziciju u nizu. Funkciju koja računa minimalni element niza i indeks implementirati kao:
void MinIndex(int *niz, int n, int *minElem, int *minIndex);,
gde je n broj elemenata.
9. Slučajnim izborom formirati niz od N elemenata iz intervala 10 do 20 (random brojevi između 10 i 20 biće elementi niza), a zatim sumirati parne elemente tog niza. Funkcija za sumiranje parnih brojeva niza treba biti napisana kao:
void SumaParnih(int *niz, int n, int *sumaParnih);
10. Napisati funkciju kojom se u stringu s izbacuje svako pojavljivanje karaktera ***ch***. Funkciju implementirati kao:
void Izbaci(char *s, char ch);
Ispisati korisniku novodobijeni string.

11. Napisati funkciju ***char *slova(char *s)*** koja vraća string u kojem su izbačeni svi karakteri iz s osim slova (cifre, znaci interpunkcije i ostalo). U glavnom programu testirati ovu funkciju zajedno sa funkcijom palindrom.
12. Napisati program koji prihvata sa standardnog ulaza pozitivan ceo broj n ($n \leq 50$), a zatim prihvata po jedan element n-dimenzionalnog niza celih brojeva. Korišćenjem funkcije formirati drugi niz koji sadrži samo ne-negativne elemente unetog niza, a potom članove drugog niza ispisati na standardni izlaz. Funkciju implementirati kao:
void izmeni(int *niz1, int n1, int niz2, int *n2)
gde su N1 i N2 dimenzije nizova Niz1 i Niz2 respektivno.
13. Napisati program u C-u koji učitava niz realnih brojeva od n elemenata i realan broj x. Napisati funkciju ***PoslednjaPojava*** koja vraća najveći indeks elemenata niza koji je jednak broju x, a ako ga nema vraća -1. Iz glavnog programa pozvati funkciju ***PoslednjaPojava***. Funkciju implementirati kao:
void PoslednjaPojava (int *Niz, int n, int *najveciIndex);
14. Napisati funkciju ***void zameni3(char *s, char *ns)*** koja zamenjuje svaki karakter stringa s sa 3 naredna karaktera po abecedi i tako dobijen string čuva u ns.
Na primer: string ANA treba modifikovati u BCDOPQBCD. Napisati potom program koji sa ulaza učitava 5 stringa maksimalne dužine 20 karaktera i ispisuje odgovarajuće modifikacije.
15. Napisati funkciju za izračunavanje aritmetičke sredine niza i testirati je.
a) aritmetička sredina je povratna vrednost funkcije
b) aritmetička sredina se vraća preko pokazivača
16. Napisati funkciju koja računa broj parnih elemenata niza na neparnim indeksima i testirati je.
a) broj traženih elemenata je povratna vrednost funkcije
b) broj traženih elemenata se vraća preko pokazivača
17.
a) Napisati funkciju kojom se elementi niza pomeraju za jednu poziciju ulevo.
Na primer: ako je dat niz **1 2 3 4 5**, posle pomeranja za poziciju ulevo niz je **2 3 4 5 1**.
Implementirati funkciju isključivo korišćenjem pokazivača, a prototip funkcije je:
void izmeni (int *Niz, int n)
b) Koristeći prethodnu funkciju, napisati program kojim se učitava dimenzija niza ($n < 50$) i elementi niza, a zatim i ceo broj k ($k > 0$) i ispisuje se niz dobijen pomeranjem elemenata za k pozicija ulevo. Implementirati funkciju isključivo korišćenjem pokazivača, a prototip funkcije je:
void izmeni (int *Niz, int n, int m)
Na primer: ako korisnik unese $n=5$, niz 3 4 5 6 7 i $k=3$ rezultat je 6 7 3 4 5

18. Napisati funkciju **`void ubaci(int a[], int* pn, int x)`** koja ubacuje element x u rastući niz a tako da niz ostane u rastućem poretku. Funkcija ažurira dimenziju niza preko pokazivača pn. Na primer: za niz 3 4 8 9 9 15 dužine 6 i element 5 novi niz je 3 4 5 8 9 9 15 i nova dužina je 7. U glavnom programu učitati dimenziju niza (n<20) i elemente niza i testirati rad funkcije.
19. Napisati funkciju **`void izbaci(int a[], int *pn, int x)`** koja iz niza a izbacuje sva pojavljivanja elementa x. Funkcija ažurira dimenziju niza preko pokazivača pn. Na primer: za niz 8 4 2 8 0 -3 dužine 6 i element 8 novi niz je 4 2 0 -3 i nova dužina je 4. U glavnom programu učitati dimenziju niza (n<20) i elemente niza i testirati rad funkcije.
20. Napisati program koji formira matricu dimenzije mxn čije elemente bira slučajnim izborom brojeva od 0 do 9. Izračunati sumu parnih elemenata, i prebrojati koliko je elemenata jednako nuli, pa zatim prikazati matricu i dobijene rezultate. Funkcija koja sumira parne brojeve, odnosno koja prebrojava koliko je elemenata jednako nuli treba biti implementirana preko pokazivača:
`void Sumiraj(int *a, int N, int *suma);`
`void JednakiNuli(int *a, int N, int *broj);`
21. Napisati funkciju kojom se na osnovu parametra koji je pokazivač na string ispituje da li je string palindrom. String je palindrom ako se isto čita s leva kao s desna. Implementirati funkciju isključivo korišćenjem pokazivača.
22. Napisati funkciju koja izbacuje sve nule na početku stringa. Argument funkcije mora biti pokazivač na string. Implementirati funkciju isključivo korišćenjem pokazivača.
23. Napisati funkciju kojom se ispisuje:
a) prvih k znakova string promenljive s
b) poslednjih k znakova string promenljive s;
c) deo stringa s od pozicije p na dužini k.
Implementirati funkciju isključivo korišćenjem pokazivača.
24. Napisati funkciju koja dati string s skraćuje na podstring od prvih p znakova. Funkcija kao argumente koristi pokazivače. Implementirati funkciju isključivo korišćenjem pokazivača.
25. Napisati funkciju koja umeće string s2 u string s1 počev od pozicije p i vraća pokazivač na s1. Implementirati funkciju isključivo korišćenjem pokazivača umesto nizova karaktera.
26. Napisati funkciju čiji je rezultat pokazivač na podstring datog stringa s koji se nalazi iza prvog pojavljivanja datog stringa s1. Implementirati funkciju isključivo korišćenjem pokazivača.
27. Napisati funkciju čiji je rezultat pokazivač na string u koji je izdvojen podstring datog stringa s koji prethodi prvom pojavljivanju datog stringa sl. Implementirati funkciju isključivo korišćenjem pokazivača.

28. Napisati funkciju koja uklanja podstring od s počev od pozicije p na dužini d i vraća pokazivač na s. Implementirati funkciju isključivo korišćenjem pokazivača.
29. Napisati funkciju kojom se u stringu s izbacuje svako pojavljivanje stringa s1. Funkcija kao rezultat vraća pokazivač na novo dobijeni string. Implementirati funkciju isključivo korišćenjem pokazivača.
30. Napisati program koji koristeći funkciju konvertuje sva velika u mala slova u učitanoj stringu i izbacuje sve cifre iz učitanoj stringu. Implementirati funkciju isključivo korišćenjem pokazivača.
31. Napisati funkciju koja će da služi za "sažimanje" stringa, tako da se svaki podniz nula zameni jednom nulom. Argument funkcije mora biti pokazivač na string. Implementirati funkciju isključivo korišćenjem pokazivača.
32. Napisati program u C-u koji učitava niz realnih brojeva od n elemenata. Napisati funkciju **TransNiz** koja transformiše učitani niz tako što zamenjuje vrednosti dva susedna elementa (prvi-drugi, treći-četvrti...), ukoliko je niz neparne dužine poslednji element ostaje nepromenjen. U glavnom delu programa ispisati dobijeni niz. Koristiti pokazivač kao argument funkcije, a ne niz. Implementirati funkciju isključivo korišćenjem pokazivača.
- 33.
- Napisati funkciju kojom se elementi niza pomeraju za jednu poziciju ulevo.
Na primer: ako je dat niz 1 2 3 4 5, posle pomeranja za jednu poziciju ulevo niz je 2 3 4 5 1. Koristiti pokazivač kao argument funkcije, a ne niz
 - Koristeći prethodnu funkciju, napisati program kojim se učitava dimenzija niza (n<50) i elementi niza, a zatim i ceo broj k (k>0) i ispisuje se niz dobijen pomeranjem elemenata za k pozicija ulevo.
Na primer: ako korisnik unese n=5, niz 3 4 5 6 7 i k=3 rezultat je 6 7 3 4 5
34. Napisati funkciju **void nalazi(int a[], int n, int x, int *rez)** kojom se proverava da li se element x nalazi u nizu a dužine n. Promenljiva *rez treba da bude -1 ukoliko se element ne nalazi u nizu tj. indeks prvog pojavljivanja elementa ako se element nalazi u nizu. Na primer:
za niz 2 3 4 5 i element 7 povratna vrednost je -1
za niz 9 4 1 8 i element 1 povratna vrednost je 2
U glavnom programu učitati dimenziju niza (n<20) i elemente niza i testirati rad funkcije.
- 35.
- Napisati funkciju **void broj_pojavljivanja(int a[], int n, int x)** koja računa broj pojavljivanja elementa x u nizu a dužine n.
 - Modifikovati prethodnu funkciju u funkciju **void broj_pojavljivanja(int a[], int n, int x, int* bp)** gde se broj pojavljivanja elementa x vraća preko pokazivača bp.
U glavnom programu učitati dimenziju niza (n<20) i elemente niza i testirati rad funkcija.

36. Napisati funkciju **`void ubaci(int a[], int* pn, int x)`** koja ubacuje element `x` u rastući niz `a` tako da niz ostane u rastućem poretku. Funkcija ažurira dimenziju niza preko pokazivača `pn`. Na primer: za niz 3 4 8 9 9 15 dužine 6 i element 5 novi niz je 3 4 5 8 9 9 15 i nova dužina je 7. U glavnom programu učitati dimenziju niza ($n < 20$) i elemente niza i testirati rad funkcije.
37. Napisati funkciju **`void izbaci(int a[], int *pn, int x)`** koja iz niza `a` izbacuje sva pojavljivanja elementa `x`. Funkcija ažurira dimenziju niza preko pokazivača `pn`. Na primer: za niz 8 4 2 8 0 -3 dužine 6 i element 8 novi niz je 4 2 0 -3 i nova dužina je 4. U glavnom programu učitati dimenziju niza ($n < 20$) i elemente niza i testirati rad funkcije.
38. 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.
39. Napisati program kojim se sa ulaza učitava ceo broj `k` ($k \leq 10$) i niska `s` maksimalne dužine 20 karaktera, a zatim se ispisuje nova niska formirana nadovezivanjem polazne stringa `n` puta. Na primer: za unos 4 i `ana` izlaz je `ana_ana_ana_ana`
40. Napisati funkciju **`int anagrami (char *s1, char *s2)`** kojom se proverava da li su dva zadate stringa anagrami (za dva stringa kažemo da su anagrami ako se od slova prvog stringa može formirati drugi stringa). Na primer: `ortoped` i `torpedo`. U glavnom programu učitati dva stringa maksimalne dužine 20 i testirati rad funkcije.