

Programiranje 2

Jednostruko povezane liste

1 ZADACI SA ČASA

Zadatak 1 Napisati biblioteku za rad sa jednostruko povezanim listama koje sadrže cele brojeve.

1. Definirati strukturu `Cvor` koja sadrži ceo broj vrednost i pokazivač na sledeći čvor liste.
2. Napisati funkciju `Cvor* napravi_cvor(int broj)`; koja kao povratnu vrednost vraća čvor čija je vrednost prosledjeni broj.
3. Napisati funkciju `void dodaj_na_pocetak(Cvor** lista, int* broj)`; koja pravi novi čvor čija je vrednost prosledjeni broj i postavlja taj čvor na početak prosledjene liste.
4. Napisati funkciju `void ispisi_listu(Cvor* lista, FILE* f)`; koja ispisuje prosledjenu listu u fajl `f`. Pogledati format ispisa u primeru ispod.
5. Testirati prve 4 stavke, tako što se sa standardno ulaza učitava broj `n`, a zatim `n` brojeva. Brojeve dodavati redom na početak liste. Listu ispisati na standardni izlaz.

Primer 1:	Primer 2:	Primer 3:
5	7	-5
1 2 3 4 5	12 10 56 -98 15 2 100	-1
[5, 4, 3, 2, 1]	[100, 2, 15, -98, 56, 10, 12]	

6. Napisati funkciju `void dodaj_na_kraj(Cvor** lista, int broj)`; koja pravi novi čvor čija je vrednost prosledjeni broj i dodaje taj čvor na kraj prosledjene liste.
7. Napisati funkciju `void ucitaj_listu(Cvor** lista, FILE* f)`; koja iz datoteke `f` učitava brojeve sve dok ne dodje do kraja datoteke i za svaki učitani broj pravi novi čvor i dodaje ga na kraj liste.
8. Napisati funkciju `void oslobodi_listu(Cvor* lista)`; koja oslobadja memoriju koju zauzima prosledjena lista.
9. Testirati napisane funkcije pozivom iz `main-a`, a učitavajući ulaz sa standardnog ulaza.

Primer 1:	Primer 2:
1 2 3 4 5	12 10 56 -98 15 2 100
[1, 2, 3, 4, 5]	[12, 10, 56, -98, 15, 2, 100]

Ova biblioteka će biti na raspolaganju studentima u svakom ispitnom roku.

Zadatak 2 Napisati funkciju `void obrisi_cvor(Cvor** lista, int broj)`; koja iz prosledjene liste brise sve čvorove čija je vrednost prosledjeni broj. Testirati funkciju pozivom iz `main-a`. Sa standardnog ulaza se učitavaju elementi liste, sve dok se ne unese 0. Zatim se unosi i broj koji treba da se obrise. Na standardni izlaz ispisati rezultujuću listu. U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: 1 2 3 4 5 0 3 [1, 2, 4, 5]	Primer 2: 1 1 2 2 3 1 1 4 4 1 1 0 1 [2, 2, 3, 4, 4]
------------------------------------------------------	---------------------------------------------------------------------

Zadatak 3 Napisati program koji objedinjuje dve sortirane liste u jednu sortiranu listu. Funkcija ne treba da kreira nove, već da samo preraspodeli postojeće čvorove. Prva lista se učitava iz datoteke čije ime se zadaje kao prvi argument komandne linije, a druga iz datoteke čije se ime zadaje kao drugi argument komandne linije. Liste u datotekama su već sortirane. Rezultujuću listu ispisati na standardni izlaz.

U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: ./a.out ulaz1.txt ulaz2.txt ulaz1.txt 1 4 8 90 999 ulaz2.txt -88 2 3 77 [-88, 1, 2, 3, 4, 8, 77, 90, 999]

Zadatak 4 Napisati funkciju `int veci_od(Cvor* lista, int broj)`; koja proverava koliko ima čvorova u listi čija je vrednost veća od prosledjenog broja. Iz datoteke koja se zadaje kao argument komandne linije se učitavaju elementi liste. Sa standardnog ulaza se učitava broj x . Na standardni izlaz ispisati rezultat izvršavanja napisane funkcije. U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: ./a.out ulaz.txt ulaz1.txt 1 4 8 90 999 4 -34 1 3 7 391 234 300 2

Zadatak 5 Napisati funkciju koja briše svaki element liste koji je manji od sume svih prethodnih elemenata u listi. Prilikom računanja sume uzeti u obzir i prethodno obrisane elemente. Kreirati glavni program koji učitava listu (elementi liste se učitavaju sve do unosa broja 0), poziva napisanu funkciju i ispisuje dobijenu listu na izlaz.

U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: 1 4 2 6 3 1 4 2 1 0 [1, 4]	Primer 2: 1 2 3 4 5 0 [1, 2, 3]	Primer 3: 8 1 3 2 7 4 3 1 4 6 2 1 0 [8]	Primer 4: 0 []
---------------------------------------------------	----------------------------------------------	------------------------------------------------------	-----------------------------

Zadatak 6 Napisati funkciju koja u datoj listi između svaka dva elementa čiji su zbir ili razlika jednaki datom broju k umeće -1. Glavni program učitava listu (sve do unosa broja 0) i ceo broj k . Potrebno je ispisati rezultujuću listu na standardni izlaz. Nije dozvoljeno korišćenje pomoćne liste. Ne analizirati prvi i poslednji element liste jer oni nemaju oba suseda.

U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: 1 2 3 1 2 3 0 3 1 -1 2 3 1 -1 2 3	Primer 2: 4 2 1 5 6 2 4 0 2 4 -1 2 1 5 6 2 -1 4	Primer 3: 1 3 1 1 3 1 0 2 1 -1 3 -1 1 -1 1 -1 3 -1 1	Primer 4: 0 5
-----------------------------------------------------------------	---------------------------------------------------------------------	--------------------------------------------------------------------------	----------------------------

Zadatak 7 Napisati funkciju koja u datoj listi izbacuje susedne elemente čiji je zbir jednak datom broju k . Potrebno je ispisati tako dobijenu listu na standardni izlaz. Nije dozvoljeno korišćenje pomoćne liste. Nije dovoljno samo ispisati traženu listu već je potrebno elemente zaista izbaciti i konstruisati novu listu. Elementi liste su celi brojevi, lista se unosi sa standardnog ulaza sve dok se ne unese 0. Nakon unosa elemenata liste unosi se broj k . U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer 1: 13 4 5 10 0 9 13 10	Primer 2: 13 4 5 9 9 0 9 13 9 9	Primer 3: 13 4 5 4 3 0 9 13 3	Primer 4: 4 5 3 -2 11 -2 11 -2 0 9 3
---------------------------------------------------	-----------------------------------------------------	---------------------------------------------------	----------------------------------------------------------

2 DOMAĆI ZADACI

Zadatak 8 Napisati funkcije za rad sa sortiranim listama:

1. void dodaj_sortirano(Cvor** lista, int broj); koja umeće broj u neopadajuće sortiranu listu, tako da lista i nakon umetanja ostane sortirana.
2. Cvor* pretrazi_sortiranu_listu(Cvor* lista, int broj); koja vraća prvi čvor sortirane liste čija je vrednost prodjedjeni broj ili NULL ukoliko takav čvor ne postoji.
3. void obrisi_sortirano(Cvor** lista, int broj); koja iz sortirane liste briše sve čvorove čija je vrednost prosledjeni broj.

Testirati funkcije pozivom iz main-a. Sa standardnog ulaza se učitavaju elementi liste, sve dok se ne unese 0. Zatim se unosi i broj koji se traži. Ukoliko je čvor sa tom vrednošću pronadjen, ispisati Ima, a ako nije ispisati Nema. Zatim se unosi broj koji treba da se obriše iz liste i na standardni izlaz treba ispisati rezultujuću listu. U slučaju greške na standardni izlaz za greške ispisati -1.

Napomena: Koristiti biblioteku za rad sa listama.

Primer1: Unesite elemente: 1 2 3 4 5 0 Trazeni element: 3 Ima Za brisanje: 3 [1, 2, 4, 5]	Primer 2: Unesite elemente: 1 1 2 2 3 3 4 4 4 5 5 0 Trazeni element: 5 Ima Za brisanje: 1 [2, 2, 3, 3, 4, 4, 5, 5]
--------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------