

PRAKTIKUM IZ PROGRAMIRANJA 2

- domaći zadatak broj 4 -

Sastaviti program na programskom jeziku C koji vrši određenu vrstu obrade nad jednom ili više **jednostruko ulančanih lista**. Program treba da:

- učita sve potrebne podatke;
- izvrši zahtevanu obradu;
- ispiše sve dobijene rezultate po tačno definisanom formatu u tekstu zadatka;

Zavisno od rednog broja problema, svaki student treba da sastavi **jedan** od programa koji su dati u prilogu ovog dokumenta.

Čvor (element) ulančane liste, treba realizovati u skladu sa zahtevima zadatka. Po potrebi, element liste može sadržati i dodatna polja u odnosu na podatke navedene u tekstu.

Voditi računa o pravilnom alociranju i dealociranju dinamičke memorije. Nakon obrade (unos, ispis unetih podataka, obrada, ispis dobijenih rezultata) dealocirati svu dinamičku memoriju. Količina memorije koja je odvojena za smeštanje podataka mora biti takva da iskorišćenje na kraju učitavanja bude potpuno (ni bajt više, ni bajt manje). U toku rada, prilikom svake alokacije dinamičke memorije, proveravati uspešnost poziva `malloc` funkcije. U slučaju neuspešne dodele dinamičke memorije, u programu ispisati poruku `MEM_GRESKA` i korektno prekinuti izvršavanje (vraćanjem vrednosti 0 kao rezultata izvršavanja programa).

Radi boljeg testiranja programa, odabrati nekoliko skupova podataka sa kojima će program biti testiran. Svaki primer treba da sadrži ulazne podatke i očekivani izlaz za te podatke.

Napomene:

- Rok za predaju trećeg domaćeg zadatka je **utorak, 24.05.2022.** putem kursa predmeta na Moodle platformi za elektronsko učenje. Tačan termin za predaju će biti naknadno definisan za sve studente. Termin će biti ograničenog vremenskog trajanja.
- Domaći zadaci će biti testirani i ocenjivani korišćenjem javnih i tajnih testova.
- Studentima će nekoliko dana pre roka za predaju biti dostupno okruženje za testiranje rešenja domaćeg zadatka na Moodle platformi za elektronsko učenje korišćenjem javnih testova.
- Prilikom predaje domaćeg zadatka studenti će rešavati i kratak test znanja u vezi rešenja domaćeg zadatka i relevantnog gradiva iz programskog jezika C koje obuhvata temu domaćeg zadatka.
- Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preduzmu odgovarajuće disciplinske mere.
- Svi drugi detalji oko predaje, ocenjivanja i odbrane domaćeg zadatka će biti blagovremeno objavljeni.
- Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa): **$i = (R + G) \bmod 5$**
- Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:
 - dz4.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

17.05.2022. godine

sa predmeta

0. Napisati program na programskom jeziku C koji obrađuje ulančanu listu koja sadrži informacije o naučnim radovima. Svaki naučni rad je određen jedinstvenim identifikacionim DOI brojem (niz znakova), naslovom rada (niz znakova), naslovom časopisa (niz znakova) u kome je rad objavljen i godinom izdanja koji su odvojeni zarezima. Potrebno je za svaki časopis koji se pojavljuje u listi formirati listu radova koji su u njemu izašli uređenu rastuće po godini izdavanja rada, a u okviru iste godine leksikografski po naslovu rada. Poredak časopisa u novoformiranoj listi treba da bude neizmenjen. DOI broj se zadaje u obliku **prefix/suffix**, gde je prefiks oblika **10.NNNN**, gde je **NNNN** najmanje četvorocifreni broj veći ili jednak 1000. Ukoliko bar jedan uneseni rad ima neispravan DOI broj potrebno je ispisati poruku **GRESKA** i korektno završiti program. Prilikom ispisa novoformirane liste radova, najpre ispisati ime časopisa, a zatim naslove u njemu objavljenih radova.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan red sa standardnog ulaza prema formatu koji je zadat u tekstu zadatka i formira novi element ulančane liste. Ukoliko pročita prazan red, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu radova. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja pravi novu ulančanu listu prema opisanom pravilu.
- 4) Implementirati funkciju koja ispisuje novoformiranu listu prema navedenom opisu.
- 5) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 6) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o radovima sa ulaza, izvrši traženu obradu i ispiše novoformiranu jednostruko ulančanu listu prema traženom formatu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
10.1002/cpe.6941,Dynamic convolutional neural network planning,CCPE,2022 10.1007/s11554-022-01218-y,Real-time double JPEG forensics,JRTIP,2021 10.1002/cpe.7048,ECG denoising method,CCPE,2020 10.1007/s11554-022-01214-2,SimpleMeshNet,JRTIP,2021 10.2298/CSIS201229045I,Entropy-based Network,ComSIS,2021	CCPE ECG denoising method Dynamic convolutional neural network planning JRTIP Real-time double JPEG forensics SimpleMeshNet ComSIS Entropy-based Network
10.1002/cpe.6941,Dynamic convolutional neural network planning,CCPE,2022 10.0007/s11554-022-01218-y,Real-time double JPEG forensics,JRTIP,2021 10.102/cpe.7048,ECG denoising method,CCPE,2020	GRESKA
10.1007/s11554-022-01214-2, SimpleMeshNet,JRTIP, 2021 10.1007/s11554-022-01218-y,Real-time double JPEG forensics,JRTIP,2021	JRTIP Real-time double JPEG forensics SimpleMeshNet

1. Napisati program na programskom jeziku C koji obrađuje ulančanu listu koja sadrži informacije o opasnim materijama koje se nalaze u okviru nekog skladišta. Svaki kontejner koji sadrži opasnu materiju je opisan jedinstvenim identifikacionim brojem, masom u kilogramima, klasom opasnih materija (ceo broj od 1 do 9) i UN brojem opasne materije (četvorocifren ceo broj). Potrebno je za svaku klasu opasnih materija formirati zasebnu listu opasnih materija uređenu opadajuće po ukupnoj masi u svim kontejnerima koji je sadrže, a u slučaju iste ukupne mase prednost ima manji UN broj. Ukoliko je UN broj neke učitane opasne materije nekorektan ispisati poruku GRESKA i korektno završiti program. Na standardni izlaz ispisati formirane ulančane liste uređene po klasama prema formatu zadatom u primerima. Ukoliko je neka klasa prazna, ne ispisivati podatke o njoj.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan red sa standardnog ulaza prema formatu koji je zadat u tekstu zadatka i formira novi element ulančane liste. Ukoliko pročita prazan red, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu sa podacima o kontejnerima sa opasnim materijama. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o kontejnerima sa opasnim materijama vrši traženu obradu.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o kontejnerima sa opasnim materijama, izvrši traženu obradu i ispiše formirane ulančane liste. Kraj unosa se označava praznim redom. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
2 1000 3 1203 4 2000 3 1202 1 5000 2 1016 7 200 2 1017 15 100 3 1090 22 10 6 1092 11 2000 3 1203 3 4000 3 1202	2: 1016 1017 3: 1202 1203 1090 6: 1092
2 1000 3 1203 4 2000 3 1202 1 5000 2 0016 7 200 2 1017 11 2000 3 1203 3 4000 3 1202	GRESKA
2 1000 3 1203 4 2000 3 1202 11 10000 3 1203 3 4000 3 1202	3: 1203 1202

2. Napisati program na programskom jeziku C koji simulira rad blagajne nekog kluba za prodaju karata za utakmicu. Blagajna sadrži spisak ljudi u vidu ulančane liste koji žele da kupe kartu. Podatak o jednom čoveku su e-mail adresa, broj karata koje želi da kupi i oznaku da li se radi o članu klubu (Y – član, N – nije član). Blagajna prodaje karte u koracima. Najpre se unese maksimalan broj raspoloživih karata. Zatim se karte raspodeljuju na sledeći način: najpre se po jedna karta podeli članovima kluba, a zatim se vrši podela preostalih karata prema traženim količinama po redosledu navedenom u ulančanoj listi. Raspodela karata se završava kada se sve karte podele ili kada na blagajni nema više karata za prodaju. Na kraju podele za svakog prijavljenog ispisati podatke i broj karata koje je dobio. Ukoliko je raspoloživ broj karata negativan broj potrebno je ispisati poruku **GRESKA** i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan red sa standardnog ulaza prema formatu koji je zadat u tekstu zadatka i formira novi element ulančane liste. Ukoliko pročita prazan red, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu ljudi koji žele da kupe kartu. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja vrši podelu karata na opisani način. Nakon oba kruga podele karata ispisati sve one koji su dobili karte u tom krugu i broj dobijenih karata.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji pročita broj raspoloživih karata sa ulaza, a potom korišćenjem prethodno realizovanih funkcija pročita podatke o osobama koje žele da kupe karte i nakon toga izvrši raspodelu karata na opisani način i ispiše one koji su dobili karte na standardnom izlazu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
10 aki@bg.com 3 Y dj@etf.rs 4 N aleksa@etf.bg.ac.rs 1 N micko@etf.bg.ac.rs 2 Y	aki@bg.com 1 micko@etf.bg.ac.rs 1 aki@bg.com 2 dj@etf.rs 4 aleksa@etf.bg.ac.rs 1 micko@etf.bg.ac.rs 1
5 aki@bg.com 3 Y dj@etf.rs 4 N aleksa@etf.bg.ac.rs 1 N micko@etf.bg.ac.rs 2 Y	aki@bg.com 1 micko@etf.bg.ac.rs 1 aki@bg.com 2 dj@etf.rs 1
-5	GRESKA

3. Napisati program na programskom jeziku C koji pomaže korisniku u organizaciji sastanaka u toku jednog dana. Ulančana lista sadrži podatke o terminima sastanaka u toku jednog dana, a svaki sastanak je definisan vremenom početka i kraja **HH:MM-HH:MM**, mestom održavanja (jedna reč) i kratkim opisom sastanka (niz znakova). Program treba da formira i ispiše ulančanu listu sastanaka na koji korisnik treba da ode u okviru zadatog dana uređenu rastuće po vremenu početka sastanaka. Ukoliko se dva ili više sastanaka preklapaju, prednost dati onome koji se odvija na istom mestu kao i prethodni, a u slučaju istog mesta održavanja, prioritet dati onom koji počinje najranije. U slučaju istog vremena održavanja, izabрати onaj koji je prvi učitani. Ukoliko je vreme početka ili kraja nekorektno, potrebno je ispisati poruku **GRESKA** i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan red sa standardnog ulaza prema formatu koji je zadat u tekstu zadatka i formira novi element ulančane liste. Ukoliko pročita prazan red, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu sa podacima o sastancima. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja ispisuje jednostruko ulančanu listu na standardni izlaz.
- 4) Implementirati funkciju koja na osnovu zadate liste sastanaka formira novu listu koja sadrži sastanke na koje korisnik treba da ode prema navedenim kriterijumima.
- 5) Implementirati funkciju koja oslobađa dinamičku memoriju koju zauzima lista koja se prosledi kao parametar funkcije.
- 6) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o sastancima korisnika, izvrši traženu obradu i ispiše formiranu ulančanu listu. Kraj unosa se označava praznim redom. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
09:00-11:00 309 Cas ASP1 08:00-08:15 37 Konsultacije 10:00-10:35 92 Sastanak K2 14:00-15:00 30 Rucak 16:00-17:00 RC Sastanak	08:00-08:15 37 Konsultacije 09:00-11:00 309 Cas ASP1 14:00-15:00 30 Rucak 16:00-17:00 RC Sastanak
09:00-11:00 309 Cas ASP1 08:00-08:15 37 Konsultacije 10:00-10:35 92 Sastanak K2 14:00-15:00 30 Rucak 15:45-17:00 RC Sastanak 16:00-17:00 30 Sastanak2	08:00-08:15 37 Konsultacije 09:00-11:00 309 Cas ASP1 14:00-15:00 30 Rucak 16:00-17:00 30 Sastanak2
09:00-25:00 309 Konferencija 08:00-08:75 37 Konsultacije 10:00-10:35 92 Sastanak K2	GRESKA

4. Napisati program na programskom jeziku C koji formira biračka mesta u nekoj opštini. Opština se sastoji od više naseljenih mesta, a svako naseljeno mesto treba da ima najmanje jedno zasebno biračko mesto. Jedno biračko mesto može da primi najviše N glasača. Podaci o glasačima se sastoje od JMBG-a (ceo broj) i mesta stanovanja. Korisnik unosi maksimalan kapacitet pojedinačnog biračkog mesta N, a program treba da rasporedi glasače po biračkim mestima u leksikografskom poretku JMBG-ova vodeći računa o naseljenom mestu u kojem žive. Kada se ispuni maksimalni kapacitet biračkog mesta (N) otvara se novo biračko mesto u toj opštini. Svakom biračkom mestu dodeliti naziv koji se sastoji od imena naseljenog mesta i rednog broja biračkog mesta. Rezultat raspoređivanja smestiti u ulančanu listu koja sadrži podatke o glasačima i dodeljnim biračkim mestima. Ukoliko je maksimalan kapacitet pojedinačnog biračkog mesta nekorektna vrednost, potrebno je ispisati poruku **GRESKA** i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan red sa standardnog ulaza prema formatu koji je zadat u tekstu zadatka i formira novi element ulančane liste. Ukoliko pročita prazan red, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu sa podacima o glasačima. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja ispisuje jednostruko ulančanu listu na standardni izlaz.
- 4) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o biračima, njihovom mestu stanovanja i maksimalnom kapacitetu biračkih mesta napravi birački spisak na opisani način.
- 5) Implementirati funkciju koja oslobađa dinamičku memoriju koju zauzima lista koja se prosledi kao parametar funkcije.
- 6) Napisati glavni program koji najpre sa standardnog ulaza učitava maksimalan kapacitet pojedinačnog biračkog mesta, a zatim korišćenjem prethodno realizovanih funkcija pročita podatke o biračima, izvrši traženu obradu i ispiše formiranu ulančanu listu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
3 5 Zvezdara 6 Kotez 9 Karaburma 11 Zvezdara 4 Zvezdara 10 Karaburma	4 Zvezdara 1 5 Zvezdara 1 6 Kotez 1 9 Karaburma 1 10 Karaburma 1 11 Zvezdara 1
3 5 Zvezdara 6 Kotez 9 Karaburma 15 Kotez 11 Zvezdara 4 Zvezdara 10 Karaburma 8 Zvezdara 1 Zvezdara	1 Zvezdara 1 4 Zvezdara 1 5 Zvezdara 1 6 Kotez 1 8 Zvezdara 2 9 Karaburma 1 10 Karaburma 1 11 Zvezdara 2 15 Kotez 1
-5	GRESKA