

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, 25.05.2021.** 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;

0. Napisati program na programskom jeziku C koji predstavlja simulaciju rešavanja „Josephus problem“-a. Određeni broj osoba (predstavljenih svojim imenima i prezimenima u vidu ulančane liste, dužina imena i prezimena nije poznata) je raspoređeno u krug i svaki korak rešavanja problema se sastoji od izbacivanja jedne osobe iz kruga, a zatim preskakanja n narednih osoba sve dok u krugu ne ostane samo jedna osoba. Pozicija osobe u krugu koja se izbacuje na samom početku i broj n se unose sa standardnog ulaza. Pozicije kreću od 0. Potrebno je prikazati stanje liste osoba nakon svakog kruga sve dok u listi ne preostane jedna osoba. Ukoliko se startna pozicija ne nalazi u listi ili je parametar n manji od 1 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 osoba. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja simulira rešenje navedenog problema. Pre svakog koraka, funkcija treba da ispiše listu u zadatom formatu. Zvezdicom se označava tekuća osoba. U koliko je lista prazna obrada se preskače.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji najpre čita startnu poziciju i parametar n , a potom korišćenjem prethodno realizovanih funkcija pročita podatke o osobama, ispiše učitane jednostruko ulančanu listu i izvrši traženu obradu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
0 2 Lacy Dyer Rosalyn Haney Madison Walton September Donaldson	Lacy Dyer Rosalyn Haney Madison Walton September Donaldson STEP0 Lacy Dyer* Rosalyn Haney Madison Walton September Donaldson STEP1 Rosalyn Haney Madison Walton* September Donaldson STEP2 Rosalyn Haney* September Donaldson STEP3 September Donaldson*
5 1 Tashya Morin Virginia Wilkinson Cleo William	GRESKA
1 1	

1. Napisati program na programskom jeziku C koji obrađuje ulančanu listu deonica koju biciklista treba da pređe. Deonica se sastoji od dužine izražene u kilometrima. Obrada se sastoji od učitavanja litara vode koje biciklista može da ponese sa sobom na početku deonice i potrošnju vode u litrama po jednom kilometru sa standardnog ulaza i prepravljanje deonica. Prepravljanje deonica se vrši od početka liste tako što se ona deonica na kojoj biciklista neće ostati bez vode ne dira, u suprotnom deonica se smanji tako da biciklista potroši svu vodu na kraju deonice. Deo deonice koji je odsečen se dodaje sledećoj deonici ukoliko to neće prouzrokovati ponovno odsecanje, u suprotnom pravi se nova deonica. Ukoliko je broj litara vode koje biciklista može da ponese ili potrošnja negativna ispisati poruku **GRESKA**.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja čita jedan realan broj sa standardnog ulaza i formira novi element ulančane liste. Ukoliko je pročitani broj manji od ili jednak 0, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu sa podacima o deonicama. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava negativnim brojem.
- 3) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o deonicama vrši traženu obradu. Nakon svake promene potrebno je ispisati celu listu.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji najpre sa standardnog ulaza čita broj litara vode koje biciklista može da ponese i potrošnju po kilometru, a potom korišćenjem prethodno realizovanih funkcija pročita podatke o deonicama, ispiše učitane ulančanu listu i izvrši traženu obradu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
5 1 7.5 8.2 1.3 4.1 11 6 -5	7.50->8.20->1.30->4.10->11.00->6.00->NULL 5.00->2.50->8.20->1.30->4.10->11.00->6.00->NULL 5.00->2.50->5.00->4.50->4.10->11.00->6.00->NULL 5.00->2.50->5.00->4.50->4.10->5.00->6.00->6.00->NULL 5.00->2.50->5.00->4.50->4.10->5.00->5.00->1.00->6.00->NULL 5.00->2.50->5.00->4.50->4.10->5.00->5.00->1.00->5.00->1.00->NULL
5 -1 20 -3	GRESKA
5 1 -1	

2. Napisati program na programskom jeziku C koji kreira pakete proizvoda. Svaki proizvod definiše njegova zapremina u kubnim centimetrima (ceo broj) i naziv, koji se može sastojati iz više reči i čija je najveća dužina 30 karaktera. Podaci o proizvodu odvojeni su jednim blanko znakom. Sa standardnog ulaza se najpre učitava zapremina paketa u kubnim centimetrima, a zatim i informacije o proizvodima, za svaki proizvod u posebnom redu. Smatrati da su svi ulazni podaci korektno zadati.

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 proizvoda. Funkcija treba da elemente ulančava u listu u poretku učitavanja. Kraj unosa se označava praznim redom.
- 3) Implementirati funkciju koja formira jedan paket proizvoda na osnovu zapremine paketa i liste učitanih proizvoda. Jedan paket proizvoda predstavljen je jednom jednostruko ulančanom listom i čini ga proizvoljan broj proizvoda, čije se dodavanje u paket pokušava od početka liste učitanih proizvoda, ali tako da proizvod može da stane u trenutni paket. Dodavanjem proizvoda u paket vrši se njegovo uklanjanje iz liste učitanih proizvoda. Proizvodi se na ovaj način dodaju u paket dok god se ne dođe do kraja liste učitanih proizvoda. Ukoliko na ovaj način nije moguće formirati neprazan paket, funkcija vraća vrednost **NULL**.
- 4) Implementirati funkciju koja ispisuje zadati paket proizvoda. Proizvodi paketa ispisuju se u pojedinačnim redovima u formatu **zapremina-naziv**. Nakon toga, u novom redu potrebno je ispisati zauzetu zapreminu paketa.
- 5) Napisati glavni program koji pročita podatke sa ulaza, a nakon toga formira jedan po jedan paket proizvoda (dok god je to moguće) i ispiše ih na standardnom izlazu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
2000 500 Jelen pivo 400 Plazma keks 1500 Ledeni caj 350 Wellness keks 300 Smoki 500 Nutella 600 Jaffa cakes	500-Jelen pivo 400-Plazma keks 350-Wellness keks 300-Smoki 1550 1500-Ledeni caj 500-Nutella 2000 600-Jaffa cakes 600
500 200 Milka cokolada 50 Twix 50 Mars 300 Cips Cacanac 150 Pardon stapici 1000 Nutella 250 Bajadera	200-Milka cokolada 50-Twix 50-Mars 150-Pardon stapici 450 300-Cips Cacanac 300 250-Bajadera 250
100 500 Jelen pivo 400 Plazma keks 1500 Ledeni caj	

3. Napisati program na programskom jeziku C koji simulira razmenu sličica između dve osobe. Jedna osoba poseduje spisak sličica za razmenu (duplikati), a druga osoba poseduje spisak sličica koje joj nedostaju. Pretpostaviti da prva osoba poseduje samo jedan duplikat svake sličice. Svaki spisak sličica se predstavlja u vidu jednostruko ulančane liste, pri čemu se svaka sličica sastoji od rednog broja u albumu (ceo broj) i naziva igrača koji se nalazi na sličici (jedna ili više reči čija je najveća dužina 30 karaktera). Na osnovu zadatih spiskova obe osobe, potrebno je izvršiti razmenu sličica koje poseduje prva osoba, a koje nedostaju drugoj osobi i formirati spisak razmenjenih sličica.

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 sličicama. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja ispisuje jednostruko ulančanu listu na standardni izlaz.
- 4) Implementirati funkciju koja na osnovu dve jednostruko ulančane liste (spisak sličica za razmenu i spisak sličica koje nedostaju) izvršava razmenu sličica i formira novu listu razmenjenih sličica.
- 5) Implementirati funkciju koja oslobađa dinamičku memoriju koju zauzima lista koja se prosledi kao parametar funkcije.
- 6) Napisati glavni program korišćenjem prethodno realizovanih funkcija. Program najpre sa standardnog ulaza čita spisak sličica za razmenu, a zatim i spisak sličica koje nedostaju. Nakon toga, program ispisuje učitane spiskove, izvršava razmenu sličica, i ispisuje spisak razmenjenih sličica. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
38 Sergio Aguero 191 Edison Cavani 191 Edison Cavani 82 Timo Werner 76 Kurt Zouma	38 Sergio Aguero 191 Edison Cavani 191 Edison Cavani 82 Timo Werner 76 Kurt Zouma 191 Edison Cavani
6 David Luiz 121 Roberto Firmino 129 Divock Origi	6 David Luiz 121 Roberto Firmino 129 Divock Origi
39 Riyad Mahrez 23 Joe Hart 99 Rui Patricio 176 Bruno Fernandes 24 Hugo Lloris 23 Joe Hart 24 Hugo Lloris	39 Riyad Mahrez 23 Joe Hart 99 Rui Patricio 176 Bruno Fernandes 24 Hugo Lloris 23 Joe Hart 24 Hugo Lloris 23 Joe Hart 24 Hugo Lloris

4. Napisati program na programskom jeziku C koji na osnovu ulančane liste turističkih atrakcija u nekom gradu, napravi novu listu, koja sadrži one atrakcije koje turista može da obiđe u okviru 8 sati, počevši od sata zadatog sa standardnog ulaza. Potrebno je pronaći sve turističke atrakcije koje je moguće posetiti u zadanom periodu, bez obzira da li se vreme u kome turista može da poseti dve atrakcije poklapa ili ne. Turistička atrakcija se sastoji od imena (jedna reč), radnog vremena (početno i krajnje vreme u satima i minutima u formatu **HH:MM-HH:MM**) i vremena potrebnog za obilazak atrakcije (izraženog u minutima).

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja učitava jedan red sa standardnog ulaza u formatu **IME HH:MM-HH:MM MIN** i formira novi element ulančane liste. Ukoliko je pročitani red prazan, funkcija treba da vrati vrednost **NULL**.
- 2) Implementirati funkciju koja pozivanjem prethodno realizovane funkcije formira jednostruko ulančanu listu sa podacima o atrakcijama. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja ispisuje jednostruko ulančanu listu na standardni izlaz.
- 4) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o atrakcijama i početnog vremena u satima napravi listu svih atrakcija koje turista može da poseti. Turista može da poseti atrakciju ukoliko je otvorena i ukoliko se neće zavoriti za vreme koje je turisti potrebno da obiđe atrakciju.
- 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 vreme kada turista dolazi u grad, a zatim korišćenjem prethodno realizovanih funkcija pročita podatke o turističkim atrakcijama. Nakon toga, program poziva funkciju koja ispiše učitavanu listu, funkciju koja napravi listu atrakcija koje turista može da poseti i funkciju koja ispiše listu atrakcija koje turista može da poseti. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
<pre>14 Ime1 14:00-15:00 30 Ime2 12:00-13:00 30 Ime3 14:45-15:00 30 Ime4 22:00-23:30 30 Ime5 12:00-17:00 120</pre>	<pre>Ime1 14:00-15:00 30 Ime2 12:00-13:00 30 Ime3 14:45-15:00 30 Ime4 22:00-23:30 30 Ime5 12:00-17:00 120 Ime1 14:00-15:00 30 Ime5 12:00-17:00 120</pre>
<pre>18 Ime1 08:00-16:00 60 Ime2 08:00-20:00 90 Ime3 17:00-18:00 60</pre>	<pre>Ime1 08:00-16:00 60 Ime2 08:00-20:00 90 Ime3 17:00-18:00 60 Ime2 08:00-20:00 90</pre>
<pre>12 Ime1 08:00-20:00 720 Ime2 12:00-15:00 30 Ime3 14:45-15:00 30 Ime4 20:00-23:30 30 Ime5 12:00-17:00 120</pre>	<pre>Ime1 08:00-20:00 720 Ime2 12:00-15:00 30 Ime3 14:45-15:00 30 Ime4 20:00-23:30 30 Ime5 12:00-17:00 120 Ime2 12:00-15:00 30 Ime5 12:00-17:00 120</pre>