

## 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 `alloc` funkcije. U slučaju neuspešne dodele dinamičke memorije, u programu ispisati poruku `MEM_GRESKA` i korektno prekinuti izvršavanje (vraćanjem vrednosti 0 kao rezultata izvršavanja programa).

#### Napomene:

- Rok za predaju trećeg domaćeg zadatka je **ponedeljak, 22.05.2023.** 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):  **$i = (R + G) \bmod 4$**
- Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:
  - dz4.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

15.05.2023. godine

*sa predmeta*

0. Napisati program na programskom jeziku C koji raspoređuje novinske članke i reklame. Program najpre treba da sa standardnog ulaza pročita novinske članke i reklame. Svaki novinski članak je predstavljen realnim brojem u opsegu (0, 1], koji predstavlja procenat prostora koji dati članak zauzima na stranici, i njegovim nazivom (string nepoznate dužine). Svaka reklama je predstavljena samo nazivom (string nepoznate dužine). Novinski članci i reklame su zadati u zasebnim redovima. Najpre se čitaju novinski članci, a potom reklame. Sekcije su razdvojene praznim redom, a kraj unosa je takođe označen praznim redom. Nakon čitanja, program treba da formira novinske stranice tako da se na stranicu uvek dodaje najveći članak koji može da stane, sve dok ima odgovarajućih članaka koji nisu raspoređeni, a zatim prostor dopunjava jednom od neraspoređenih reklama. Ukoliko više novinskih članaka zadovoljava navedeni uslov, dodavati ih po redosledu pojavljivanja na standardnom ulazu. Novu stranicu formirati kada postojeća ne može da se dopuni preostalim člancima. Ukoliko nije unet dovoljan broj reklama, na stranicama dodati generičku poruku. Na kraju rada programa, neophodno je ispisati formirane stranice i neraspoređene reklame. Ukoliko je procentualni udeo stranice koji članak zauzima van navedenog opsega potrebno je ispisati poruku **ERROR** i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja formira jednostruko ulančanu listu novinskih članaka. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja formira jednostruko ulančanu listu reklama. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja formira i ispisuje stranice novina.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o novinskim člancima i reklamama, izvrši traženu obradu i ispiše rezultat prema traženom formatu. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

### Primeri

Ulaz:	Izlaz:
0.34 Dolorem numquam sed adipisci 0.34 Ipsum est etincidunt tempora 0.25 Voluptatem dolore dolorem velit 0.36 Quaerat tempora dolor ut dolor  Voluptatem ut quaerat Tempora sit sit ut sed Non numquam sed Velit quisquam adipisci dolor	PAGE 0 ARTICLE: Quaerat tempora dolor ut dolor ARTICLE: Dolorem numquam sed adipisci ARTICLE: Voluptatem dolore dolorem velit ADVERTISEMENT: Voluptatem ut quaerat PAGE 1 ARTICLE: Ipsum est etincidunt tempora ADVERTISEMENT: Tempora sit sit ut sed UNUSED ADVERTISEMENT Non numquam sed Velit quisquam adipisci dolor
0.36 Velit consectetur non consectetur ut. 0.23 Voluptatem aliquam tempora. 0.59 Quisquam magnam aliquam.  Eius amet quaerat est velit.	PAGE 0 ARTICLE: Quisquam magnam aliquam. ARTICLE: Velit consectetur non consectetur ut. ADVERTISEMENT: Eius amet quaerat est velit. PAGE 1 ARTICLE: Voluptatem aliquam tempora. <ADVERTISEMENT_PLACEHOLDER>
1.21 Labore eius consectetur.  Velit etincidunt neque.	ERROR

1. Napisati program na programskom jeziku C koji raspoređuje posetioce koncerta po kategorijama. Program treba najpre da pročita podatke o kategorijama i posetiocima. Za svaku kategoriju se unosi njeno ime (string nepoznate dužine), cena ulaznice (realan broj) i broj raspoloživih mesta (ceo broj). Za svakog posetioca se unosi njegovi ime i prezime (stringovi nepoznate dužine) i dve željene kategorije po prioritetima (stringovi nepoznate dužine). Najpre se čitaju podaci o kategorijama, a potom podaci o posetiocima. Sekcije su razdvojene praznim redom, a kraj unosa je takođe označen praznim redom. Nakon čitanja, program treba da rasporedi posetioce po kategorijama i dodeli im karte prema željama. Karta se dodeljuje posetiocu ukoliko ima slobodnih mesta u nekoj od željenih kategorija. Dodeljivanje se vrši po redosledu navođenja posetioca na standardnom ulazu. Ukoliko nije moguće dodeliti kartu prema željama, dodeliti kartu iz neke od preostalih kategorija koja nije skuplja od prve želje ukoliko takva postoji, ali tek nakon što se svim posetiocima dodeli karta po prvoj ili drugoj želji. Na kraju je potrebno ispisati ukupnu zaradu, dodeljene karte i imena posetioca kojima nije dodeljena karta. Ukoliko je posetilac tražio kategoriju koja ne postoji potrebno je ispisati poruku `ERROR` i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja formira jednostruko ulančanu listu sa podacima o kategorijama. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja formira jednostruko ulančanu listu sa podacima o posetiocima. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja raspoređuje karte posetiocima.
- 4) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o kategorijama i posetiocima, izvrši traženu obradu i ispiše rezultat obrade. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

## Primeri

Ulaz:	Izlaz:
<pre> vip 2000 3 stands 1500 9 floor0 1000 9 floor1 750 2 pit 500 2  Jackqueline Gregorin vip stands Ferdinanda Haney vip floor1 Jaquenetta McGuinley vip floor1 Kliment Newtown vip floor1 Kaia Loding floor0 pit </pre>	<pre> PROFIT = 7750.00 Jackqueline Gregorin -&gt; vip Ferdinanda Haney -&gt; vip Jaquenetta McGuinley -&gt; vip Kliment Newtown -&gt; floor1 Kaia Loding -&gt; floor0 </pre>
<pre> vip 2000 1 stands 1500 1 floor0 1000 1 floor1 750 2  Jackqueline Gregorin stands floor0 Ferdinanda Haney vip stands Jaquenetta McGuinley floor1 stands Kliment Newtown floor0 floor1 Kaia Loding floor0 stands Ogden January vip floor1 </pre>	<pre> PROFIT = 6000.00 Jackqueline Gregorin -&gt; stands Ferdinanda Haney -&gt; vip Jaquenetta McGuinley -&gt; floor1 Kliment Newtown -&gt; floor0 Ogden January -&gt; floor1 NO TICKETS FOR: Kaia Loding </pre>
<pre> vip 2000 63 stands 1500 58 floor0 1000 53 floor1 750 95 pit 500 82  Jackqueline Gregorin vip floor5 </pre>	<pre> ERROR </pre>

2. Napisati program na programskom jeziku C koji pomaže korisniku prilikom pravljenja rasporeda korišćenja sala. Program najpre treba da učitava podatke o nepoznatom broju sala. Svaka sala je zadata u zasebnom redu i predstavljena je imenom (string nepoznate dužine) i periodom u kome je slobodna (u formatu `hh:mm-hh:mm`). Program najpre treba da učitava informacije o salama, a potom da pročita spisak zahteva korisnika. Zahtevi su dati u istom formatu. Zahtev može da se prihvati ili odbije u zavisnosti od toga da li je sala slobodna u traženom periodu. Nakon obrade svakog zahteva neophodno je ispisati novo stanje zauzeća sala. Kraj unosa prilikom čitanja informacija o salama i prilikom čitanja zahteva je predstavljen praznim redom. Ukoliko se u inicijalnom rasporedu sala nađe vremenski opseg koji je nevalidan (sati ili minuti imaju vrednost veću od 24, odnosno 60) potrebno je ispisati poruku `ERROR` i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja formira jednostruko ulančanu listu sala i termina u kojima su one slobodne. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja obrađuje jedan zahtev korisnika.
- 3) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 4) Napisati glavni program koji prvo pročita sve informacije o salama, a potom obrađuje zahteve korisnika. Nakon obrade svakog zahteva program treba da ispiše trenutno stanje. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

### Primeri

Ulaz:	Izlaz:
59 08:00 22:00 61 08:00 22:00 62 08:00 22:00 65 08:00 22:00  59 08:00 10:00 59 12:00 15:00 61 08:00 10:00 62 08:00 10:00	REQUEST FOR 59, 08:00-10:00 APPROVED 59 10:00-22:00 61 08:00-22:00 62 08:00-22:00 65 08:00-22:00  REQUEST FOR 59, 12:00-15:00 APPROVED 59 10:00-12:00 59 15:00-22:00 61 08:00-22:00 62 08:00-22:00 65 08:00-22:00  REQUEST FOR 61, 08:00-10:00 APPROVED 59 10:00-12:00 59 15:00-22:00 61 10:00-22:00 62 08:00-22:00 65 08:00-22:00  REQUEST FOR 62, 08:00-10:00 APPROVED 59 10:00-12:00 59 15:00-22:00 61 10:00-22:00 62 10:00-22:00 65 08:00-22:00
59 08:00 22:00 61 18:00 22:00 62 08:00 22:00 65 08:00 12:00  59 08:00 10:00 59 08:00 09:00 61 08:00 10:00 61 18:17 19:07 59 15:30 16:00 59 09:30 16:00 62 08:13 10:21	REQUEST FOR 59, 08:00-10:00 APPROVED 59 10:00-22:00 61 18:00-22:00 62 08:00-22:00 65 08:00-12:00  REQUEST FOR 59, 08:00-09:00 DENIED 59 10:00-22:00 61 18:00-22:00 62 08:00-22:00 65 08:00-12:00  REQUEST FOR 61, 08:00-10:00 DENIED 59 10:00-22:00 61 18:00-22:00 62 08:00-22:00

	<p>65 08:00-12:00  REQUEST FOR 61, 18:17-19:07 APPROVED  59 10:00-22:00  61 18:00-18:17  61 19:07-22:00  62 08:00-22:00  65 08:00-12:00  REQUEST FOR 59, 15:30-16:00 APPROVED  59 10:00-15:30  59 16:00-22:00  61 18:00-18:17  61 19:07-22:00  62 08:00-22:00  65 08:00-12:00  REQUEST FOR 59, 09:30-16:00 DENIED  59 10:00-15:30  59 16:00-22:00  61 18:00-18:17  61 19:07-22:00  62 08:00-22:00  65 08:00-12:00  REQUEST FOR 62, 08:13-10:21 APPROVED  59 10:00-15:30  59 16:00-22:00  61 18:00-18:17  61 19:07-22:00  62 08:00-08:13  62 10:21-22:00  65 08:00-12:00</p>
<p>59 29:00 22:00  61 08:00 22:00  62 08:00 22:00  65 08:00 22:00    59 08:00 10:00</p>	<p>ERROR</p>

3. Napisati program na programskom jeziku C koji simulira operaciju unije skupova. Program najpre treba da pročita inicijalno stanje skupa koje je predstavljeno intervalima brojeva. Svaki interval je dat u zasebno redu i zadat je pomoću celih brojeva koji predstavljaju početak i kraj intervala. Nakon toga, program treba da sa standardnog ulaza pročita intervale koje je neophodno dodati u skup. Svaki interval je zadat u zasebnom redu i u istom formatu kao i inicijalni intervali. Prilikom dodavanja novog intervala neophodno je izvršiti sažimanje skupa ukoliko je to moguće. Kraj unosa prilikom čitanja intervala koji predstavljaju inicijalno stanje skupa i intervala koje treba dodati u skup je predstavljen praznim redom. Ukoliko je početak ili kraj nekog od intervala koji predstavljaju inicijalno stanje izostavljen, potrebno je ispisati poruku **ERROR** i korektno završiti program. Pretpostaviti da se intervali koji predstavljaju inicijalno stanje neće preklapati.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja formira jednostruko ulančanu listu intervala koji predstavljaju inicijalno stanje skupa. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja dodaje interval u skup i sažima isti ukoliko je to moguće.
- 3) Implementirati funkciju koja oslobađa dinamičku memoriju.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita inicijalno stanje skupa i dodaje nove intervale u isti. Potrebno je ispisati novo stanje skupa nakon svakog dodavanja i to tako da su intervali sortirani rastuće po početku intervala. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

### Primeri

Ulaz:	Izlaz:
7-10 12-24 29-36  54-80 35-45 90-99 99-120 39-80	ADD 54-80 7-10, 12-24, 29-36, 54-80 ADD 35-45 7-10, 12-24, 29-45, 54-80 ADD 90-99 7-10, 12-24, 29-45, 54-80, 90-99 ADD 99-120 7-10, 12-24, 29-45, 54-80, 90-120 ADD 39-80 7-10, 12-24, 29-80, 90-120
20-32 49-51 53-77  41-47 49-67 81-89 79-81 60-77 94-96 56-64 72-99	ADD 41-47 20-32, 41-47, 49-51, 53-77 ADD 49-67 20-32, 41-47, 49-77 ADD 81-89 20-32, 41-47, 49-77, 81-89 ADD 79-81 20-32, 41-47, 49-77, 79-89 ADD 60-77 20-32, 41-47, 49-77, 79-89 ADD 94-96 20-32, 41-47, 49-77, 79-89, 94-96 ADD 56-64 20-32, 41-47, 49-77, 79-89, 94-96 ADD 72-99 20-32, 41-47, 49-99
0-32 -51 53-77  41-47 49-67 5-89	ERROR