

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 i izvrši njihovo ispisivanje;
- 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 formatu ispisa definisanom u tekstu zadatka i priloženim primerima. Ukoliko korisnik unese praznu listu, ne ispisivati ništa.

Navedene korake u izvršavanju programa realizovati kao zasebne funkcije prema rasporedu navedenom u tekstu svakog od zadataka. Po potrebi, mogu se realizovati i druge, pomoćne funkcije koje vrše deo obrade. Student sam treba da definiše imena funkcija, kao i argumente potrebne za njihov rad. Voditi računa da se funkcijama dostave samo neophodni podaci (pokazivač na početak liste i, po potrebi, podatke od kojih zavisi obrada). **Potprogrami ne smeju komunicirati pomoću globalnih promenljivih, već samo preko liste argumenata i povratne vrednosti.**

Voditi računa o pravilnom alociranju i dealociranju dinamičke memorije. Nakon obrade (unos, ispis unetih podataka, obrada, ispis dobijenih rezultata) dealocirati svu dinamičku memoriju. 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).

Ako nešto u postavci zadatka nije dovoljno precizno definisano ili ako su neki od zahteva međusobno suprotstavljeni, usvojiti razumnu pretpostavku i rešiti zadatak korišćenjem te pretpostavke. Programski kod rešenja zadatka treba da bude uredno komentaran, tako da pri pregledu programa lako može biti uočeno šta radi bilo koja programska celina. Takođe, treba poštovati pravila nazublivanja (indentacije) određenih celina prilikom pisanja koda.

Radi boljeg testiranja programa, odabрати 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 četvrtog domaćeg zadatka je **sreda, 20.05.2020.** 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 i ocenjivanja domaćeg zadatka će biti blagovremeno objavljeni.

7. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa): **$i = (R + G) \bmod 6$**
8. Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:
 - **dz4.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

13.05.2020. godine

sa predmeta

0. Napisati program koji pomaže korisniku u pravljenju rasporeda časova za jedan dan pri slušanju predavanja na daljinu. Ulančana lista sadrži podatke o terminima predavanja za dati dan. Predavanje je određeno šifrom predmeta (niz od najviše 30 znakova), platformom (niz od najviše 30 znakova), vremenom početka predavanja (hh:mm), kao i trajanjem predavanja u minutima (ceo broj). Program najpre treba da učitava podatke u ulančanu listu. Korisnik zatim sa standardnog ulaza unosi vremenski interval (vreme početka i vreme kraja) i platformu za koju se pravi raspored časova. Program treba da formira i ispiše ulančanu listu predavanja koja su se našla u rasporedu časova i ispiše šifre časova koji se nisu našli u rasporedu, a održavaju se u zatom vremenskom intervalu. Ako se dva časa predavanja preklapaju, prioritet dati onom koji počinje kasnije, odnosno onom koji je prvi učitavan u listu, ukoliko počinju u isto vreme. Pretpostaviti da postoji samo jedan termin za svaki predmet i da se svaki čas preklapa sa najviše jednim časom.

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 terminima predavanja za odgovarajući dan. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja proverava da li se dva termina predavanja preklapaju.
- 4) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o terminima predavanja za odgovarajući dan i prosleđenog vremenskog intervala i platforme formira raspored časova. Šifre časova koji se nisu našli u rasporedu, a održavaju se u zatom vremenskom intervalu ispisati na standardnom izlazu, po poretku učitavanja.
- 5) Implementirati funkciju koja ispisuje jednostruko ulančanu listu sa terminima časova. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red (osim ukoliko je lista prazna). Ispis elementa liste uraditi po istom formatu kao što je ulazni.
- 6) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 7) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o terminima predavanja za dati dan i vremenski interval i platformu, zatim ispiše učitavanu ulančanu listu, formira raspored časova prema unetim podacima i ispiše poruku **Raspored:** i rezultate. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120 08:00 14:00 MS Teams	13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120 Raspored: 13S111P2, MS Teams, 09:00, 60 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120 13S111PP2
13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120	13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120

16:00 20:00 Zoom	
13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120 12:00 20:00 Zoom	13S111P2, MS Teams, 09:00, 60 13S111PP2, MS Teams, 08:00, 90 13E111ORT1, Zoom, 14:00, 120 13E081MM2, Zoom, 14:00, 180 13S111ASP1, MS Teams, 12:00, 120 13S081M2, MS Teams, 10:00, 120 Raspored: 13E111ORT1, Zoom, 14:00, 120 13E081MM2

1. Napisati program koji raspoređuje studente iz ulančane liste u zavisnosti od obaveze koju polaže u tekućem roku. Podaci o studentu se sastoje od imena i prezimena (niz od najviše 30 znakova), broja indeksa (**bbbb/gg**), osvojenom broju poena na kolokvijumu (ceo broj) i obaveze za koju se prijavio da polaže. Moguće je prijaviti se za polaganje nadoknade kolokvijuma (**NK**), ispita (**I**) ili nadoknade kolokvijuma i ispita zajedno (**NKI**). Korisnik sa standardnog ulaza unosi broj mesta za polaganje nadoknade kolokvijuma i broj mesta za polaganje nadoknade kolokvijuma i ispita zajedno, dok je broj mesta za polaganje ispita neograničen. Program treba da formira i ispiše tri ulančane liste sa studentima koji su raspoređeni za date obaveze, sortirane po indeksima. Prvo se biraju studenti koji polažu integralno (nadoknada kolokvijuma i ispit), pri čemu prednost imaju studenti sa manjim brojem indeksa. Neraspoređeni studenti se zajedno sa ostalim studentima zatim razmatraju za polaganje nadoknade kolokvijuma pri čemu imaju prednost studenti sa manjim brojem poena osvojenim na kolokvijumu. Preostali studenti polažu ispit.

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 studentima. Funkcija treba da elemente ulančava u listu u **obrnutom** poretku od poretka učitavanja.
- 3) Implementirati funkciju koja poredi studente po indeksu.
- 4) Implementirati funkciju koja poredi studente po broju osvojenih poena na kolokvijumu.
- 5) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o studentima, broja mesta za polaganje nadoknade kolokvijuma i broja mesta za polaganje nadoknade kolokvijuma i ispita zajedno formira raspored studenata po obavezama koje polažu.
- 6) Implementirati funkciju koja ispisuje jednostruko ulančanu listu sa rasporedom studenata po obavezama koju polažu. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red (osim ukoliko je lista prazna). Ispis elementa liste uraditi po indeksima.
- 7) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 8) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o studentima, broj mesta za polaganje nadoknade kolokvijuma i broj mesta za polaganje nadoknade kolokvijuma i ispita zajedno, ispiše učitavanu ulančanu listu, formira raspored studenata prema obavezama koje polažu i ispiše poruku **Raspored:** i rezultate. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
Marko Markovic, 0401/19, 51, NKI Jovan Jovanovic, 0101/19, 91, I Jelena Jelenic, 0551/18, 47, NKI Lidija Lidajic, 0210/19, 63, NKI Jana Janic, 0102/19, 76, I Petra Petrovic, 0052/19, 93, I Ognjen Ognjenic, 0212/19, 58, NK	Ognjen Ognjenic, 0212/19, 58, NK Petra Petrovic, 0052/19, 93, I Jana Janic, 0102/19, 76, I Lidija Lidajic, 0210/19, 63, NKI Jelena Jelenic, 0551/18, 47, NKI Jovan Jovanovic, 0101/19, 91, I Marko Markovic, 0401/19, 51, NKI Raspored: Jelena Jelenic, 0551/18, 47, NKI Lidija Lidajic, 0210/19, 63, NKI Ognjen Ognjenic, 0212/19, 58, NK Marko Markovic, 0401/19, 51, NK Petra Petrovic, 0052/19, 93, I Jovan Jovanovic, 0101/19, 91, I Jana Janic, 0102/19, 76, I
2	
3	

<p>Petra Petrovic, 0052/19, 93, I Ognjen Ognjenic, 0212/19, 58, NKI Marko Markovic, 0401/19, 51, NKI Jovan Jovanovic, 0101/19, 91, I Jelena Jelenic, 0551/18, 47, NKI Jana Janic, 0102/19, 76, I Lidija Lidajic, 0210/19, 63, NKI</p> <p>2 3</p>	<p>Lidija Lidajic, 0210/19, 63, NKI Jana Janic, 0102/19, 76, I Jelena Jelenic, 0551/18, 47, NKI Jovan Jovanovic, 0101/19, 91, I Marko Markovic, 0401/19, 51, NKI Ognjen Ognjenic, 0212/19, 58, NKI Petra Petrovic, 0052/19, 93, I</p> <p>Raspored:</p> <p>Jelena Jelenic, 0551/18, 47, NKI Lidija Lidajic, 0210/19, 63, NKI Ognjen Ognjenic, 0212/19, 58, NK Marko Markovic, 0401/19, 51, NK Petra Petrovic, 0052/19, 93, I Jovan Jovanovic, 0101/19, 91, I Jana Janic, 0102/19, 76, I</p>
<p>Petra Petrovic, 0052/19, 93, NKI Ognjen Ognjenic, 0212/19, 58, NKI Marko Markovic, 0401/19, 51, NKI Jovan Jovanovic, 0101/19, 91, NKI Jelena Jelenic, 0551/18, 47, NKI Jana Janic, 0102/19, 76, NKI Lidija Lidajic, 0210/18, 63, NKI</p> <p>0 3</p>	<p>Lidija Lidajic, 0210/18, 63, NKI Jana Janic, 0102/19, 76, NKI Jelena Jelenic, 0551/18, 47, NKI Jovan Jovanovic, 0101/19, 91, NKI Marko Markovic, 0401/19, 51, NKI Ognjen Ognjenic, 0212/19, 58, NKI Petra Petrovic, 0052/19, 93, NKI</p> <p>Raspored:</p> <p>Jelena Jelenic, 0551/18, 47, NK Ognjen Ognjenic, 0212/19, 58, NK Marko Markovic, 0401/19, 51, NK Lidija Lidajic, 0210/18, 63, I Petra Petrovic, 0052/19, 93, I Jovan Jovanovic, 0101/19, 91, I Jana Janic, 0102/19, 76, I</p>

2. Napisati program koji obrađuje ulančanu listu, koja sadrži informacije o rezultatima domaćeg zadatka rađenog na platformi Moodle. Za svakog studenta se beleži broj indeksa (*gggg/bbbb*), ime i prezime (niz od najviše 30 znakova) i broj osvojenih poena na domaćem zadatku (ceo broj). Postoji veći broj zadataka pri čemu svaki student rešava jedan zadatak određen formulom $(R + G) \bmod X$, gde je R redni broj indeksa, G poslednje dve cifre godine upisa, dok x unosi korisnik sa standardnog ulaza. Program treba da formira i ispiše liste rezultata za svaki zadatak i ispiše prosečan broj osvojenih poena za sve studente. Rezultate urediti leksikografski po imenu i prezimenu studenata.

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 rezultatima domaćih zadataka. Funkcija treba da elemente ulančava u listu u **obrnutom** poretku od poretka učitavanja.
- 3) Implementirati funkciju koja određuje redni broj zadatka za zadati broj indeksa u zavisnosti od ukupnog broja zadataka (x u gornjoj formuli).
- 4) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa rezultatima domaćeg zadatka i rednog broja zadataka formira ulančanu listu rezultata za prosleđeni zadatak. U okviru liste rezultate urediti leksikografski po imenu i prezimenu studenata.
- 5) Implementirati funkciju koja ispisuje prosleđenu jednostruko ulančanu listu sa rezultatima domaćeg zadatka. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red. Ispis elementa liste uraditi po istom formatu kao što je ulazni, sa dodatkom rednog broja zadatka koji prethodni ostalim podacima elementa.
- 6) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 7) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o rezultatima domaćeg zadatka i ukupan broj zadataka, ispiše učitane ulančanu listu, formira pojedinačne liste za svaki zadatak, izračuna prosečan broj osvojenih poena i ispiše rezultate. Liste ispisivati u rastućem poretku rednog broja zadatka. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
0401/2019, Marko Markovic, 51 0101/2019, Jovan Jovanovic, 91 0210/2019, Lidija Lidajic, 63 0102/2019, Jana Janic, 76 0052/2019, Petra Petrovic, 93 0212/2019, Ognjen Ognjenic, 58 6	0212/2019, Ognjen Ognjenic, 58 0052/2019, Petra Petrovic, 93 0102/2019, Jana Janic, 76 0210/2019, Lidija Lidajic, 63 0101/2019, Jovan Jovanovic, 91 0401/2019, Marko Markovic, 51 0, 0101/2019, Jovan Jovanovic, 91 0, 0401/2019, Marko Markovic, 51 1, 0102/2019, Jana Janic, 76 1, 0210/2019, Lidija Lidajic, 63 3, 0212/2019, Ognjen Ognjenic, 58 5, 0052/2019, Petra Petrovic, 93 72.00
0212/2019, Marko Rognjenic, 58 0101/2019, Jovan Covanovic, 91 0210/2019, Oliver Ridajic, 63 0401/2019, Janko Narkovic, 51 0102/2019, Jana Panic, 76 0052/2019, Marko Metrovic, 93 5	0052/2019, Marko Metrovic, 93 0102/2019, Jana Panic, 76 0401/2019, Janko Narkovic, 51 0210/2019, Oliver Ridajic, 63 0101/2019, Jovan Covanovic, 91 0212/2019, Marko Rognjenic, 58 0, 0401/2019, Janko Narkovic, 51 0, 0101/2019, Jovan Covanovic, 91 1, 0102/2019, Jana Panic, 76

	1, 0052/2019, Marko Metrovic, 93 1, 0212/2019, Marko Rognjenic, 58 4, 0210/2019, Oliver Ridajic, 63 72.00
0212/2019, Marko Rognjenic, 58 0101/2019, Jovan Covanovic, 91 0210/2019, Oliver Ridajic, 63 0401/2019, Janko Narkovic, 51 2	0401/2019, Janko Narkovic, 51 0210/2019, Oliver Ridajic, 63 0101/2019, Jovan Covanovic, 91 0212/2019, Marko Rognjenic, 58 0, 0401/2019, Janko Narkovic, 51 0, 0101/2019, Jovan Covanovic, 91 1, 0212/2019, Marko Rognjenic, 58 1, 0210/2019, Oliver Ridajic, 63 65.75

3. Napisati program koji obrađuje podatke o uslugama krojačke radnje. Ulančana lista sadrži podatke o krojačkim uslugama u toku jedne godine, koje čini ime mušterije, vrsta usluge (niz od najviše 30 znakova), status usluge (0 – započeto, 1 - završeno), datum početka izrade (dd.mm.), kao i rok izrade u danima (ceo broj). Program treba da najpre formira, a zatim ispiše ulančanu listu onih krojačkih usluga koje su započete i koje treba da budu završene do zadatog datuma (uključujući i taj datum), sortirane u neopadajućem poretku po krajnjem datumu roka. Prednost imaju krojačke usluge koje zahtevaju kraće vreme izrade.

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 krojačkim uslugama za jednu godinu. Funkcija treba da elemente ulančava u listu u **obrnutom** poretku od poretka učitavanja.
- 3) Implementirati funkciju koja proverava da li krojačka usluga treba da bude završena do zadatog datuma, uključujući i taj datum.
- 4) Implementirati funkciju koja na osnovu jednostruko ulančane liste sa podacima o krojačkim uslugama formira jednostruko ulančanu listu svih onih usluga koje su započete i koje treba da budu završene do zadatog datuma (uključujući i taj datum), sortiranu u neopadajućem poretku po krajnjem datumu roka i vremenu izrade, tim redom. Ukoliko su oba kriterijuma ista, uzeti u obzir poredak učitavanja.
- 5) Implementirati funkciju koja ispisuje jednostruko ulančanu listu sa podacima o krojačkim uslugama. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red. Ispis elementa liste uraditi po istom formatu kao što je i ulazni.
- 6) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 7) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita sve podatke o krojačkim uslugama, kao i datum završetka usluga, zatim ispiše učitavanu ulančanu listu, formira sortiranu listu započetih usluga koje treba da budu završene do zadatog datuma prema unetim podacima i ispiše rezultate. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz:	Izlaz:
Vlada, popravka slica, 0, 07.05., 1 Vida, bluza, 1, 10.05., 4 Dalija, suknja, 0, 10.05., 3 Isak, skracenje pantalona, 0, 06.05., 3 09.05.	Isak, skracenje pantalona, 0, 06.05., 3 Dalija, suknja, 0, 10.05., 3 Vida, bluza, 1, 10.05., 4 Vlada, popravka slica, 0, 07.05., 1 Vlada, popravka slica, 0, 07.05., 1 Isak, skracenje pantalona, 0, 06.05., 3
Nebojsa, pantalone, 0, 26.05., 5 Nemanja, skracenje rukava, 1, 11.06., 3 Marta, haljina, 0, 25.06., 10 Petra, haljina, 0, 15.06., 7 01.07.	Petra, haljina, 0, 15.06., 7 Marta, haljina, 0, 25.06., 10 Nemanja, skracenje rukava, 1, 11.06., 3 Nebojsa, pantalone, 0, 26.05., 5 Nebojsa, pantalone, 0, 26.05., 5 Petra, haljina, 0, 15.06., 7
Petra, haljina, 0, 25.05., 7 Petra, suknja, 0, 25.05., 7 01.06.	Petra, suknja, 0, 25.05., 7 Petra, haljina, 0, 25.05., 7 Petra, suknja, 0, 25.05., 7 Petra, haljina, 0, 25.05., 7

4. Napisati program koji pomaže organizatorima konferencije da sastave program dešavanja. Ulančana lista sadrži podatke o događajima na konferenciji, koju čini njegova tema (niz od najviše 50 znakova), dužina trajanja (u minutima), kao i sala u kojoj se dešava (pozitivan ceo broj), u toku jednog dana. Program treba najpre da sortira događaje redom po sali i dužini trajanja u neopadajućem poretku, zatim da na osnovu zadatog vremena početka konferencije (**hh:mm**) izračuna početak i kraj svakom događaju u listi, kao i da na kraju ispiše kompletan program dešavanja konferencije. Smatrati da događaji uvek mogu da se smeste u okviru jednog dana u odnosu na zadati početak konferencije.

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 događajima u toku dana. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja sortira zadatu listu događaja po sali i po dužini trajanja, redom. Ukoliko su oba kriterijuma ista, uzeti u obzir poredak učitavanja.
- 4) Implementirati funkciju koja svakom događaju u listi izračuna početak i kraj, na osnovu zadatog vremena početka konferencije. Događajima dodeljivate vreme početka i kraja u poretku u kome se nalaze u prosleđenoj listi.
- 5) Implementirati funkciju koja ispisuje jednostruko ulančanu listu sa podacima o događajima na konferenciji, koji imaju vreme početka i kraja. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red. Ispis elementa liste uraditi po formatu koji je dat u primerima izlaza.
- 6) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 7) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o događajima konferencije u toku jednog dana, kao i sam početak konferencije, ispiše učitane ulančanu listu, formira sortiranu listu događaja konferencije sa podacima o početku i kraju događaja i ispiše rezultate. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz	Izlaz
ROS use cases in robotics, 60, 65 Open-source business model, 45, 56 Data for the future, 60, 62 Free software, 45, 65 09:00	ROS use cases in robotics, 60, 65 Open-source business model, 45, 56 Data for the future, 60, 62 Free software, 45, 65 Open-source business model, 56, 9:00-9:45 Data for the future, 62, 9:45-10:45 Free software, 65, 10:45-11:30 ROS use cases in robotics, 65, 11:30-12:30
ROS use cases in robotics, 60, 65 Data for the future, 55, 62 Clinical engineering, 60, 65 Ada Byron contributions, 45, 56 Open-source business model, 60, 56 10:00	ROS use cases in robotics, 60, 65 Data for the future, 55, 62 Clinical engineering, 60, 65 Ada Byron contributions, 45, 56 Open-source business model, 60, 56 Ada Byron contributions, 56, 10:00-10:45 Open-source business model, 56, 10:45-11:45 Data for the future, 62, 11:45-12:40 ROS use cases in robotics, 65, 12:40-13:40 Clinical engineering, 65, 13:40-14:40
Open-source business model, 30, 56 Data for the future, 30, 56 Free software, 30, 65 12:00	Open-source business model, 30, 56 Data for the future, 30, 56 Free software, 30, 65 Open-source business model, 56, 12:00-12:30 Data for the future, 56, 12:30-13:00 Free software, 65, 13:00-13:30

5. Napisati program koji obrađuje podatke korisnika sportske aplikacije za trčanje; ulančana lista sadrži podatke o aktivnostima korisnika u toku jedne godine, koju čini trajanje aktivnosti u minutima (ceo broj), broj pređenih kilometara (realan broj) i datum aktivnosti (dd/mm). Program treba da ažurira i ispiše listu aktivnosti, koja će imati sumirane informacije o svim aktivnostima iz istoga dana, zajedno sa prosečnom brzinom trčanja (km/h), zaokruženom na dve decimale.

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 aktivnostima u toku godine. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 3) Implementirati funkciju koja sortira zadatu listu aktivnosti po datumima.
- 4) Implementirati funkciju koja sažima zadatu listu aktivnosti tako što sve aktivnosti iz istog dana sumira u jednu aktivnost, koja treba da sadrži ukupan broj minuta i pređenih kilometara, kao i novi podatak o prosečnoj brzini trčanja za taj dan. Funkcija treba da zadrži relativan poredak sažetih čvorova, kao u prosleđenoj listi.
- 5) Implementirati funkciju koja ispisuje jednostruko ulančanu listu sa sumiranim podacima o aktivnostima u toku godine, koji uključuju i prosečnu brzinu trčanja. Svaki element liste treba ispisati u zasebnom redu. Nakon poslednjeg elementa liste ispisati znak za novi red. Ispis elementa liste uraditi po formatu koji je dat u primerima izlaza.
- 6) Implementirati funkciju koja oslobađa svu zauzetu dinamičku memoriju.
- 7) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita podatke o aktivnostima u toku jedne godine, ispiše učitane ulančanu listu, formira sortiranu listu aktivnosti sa sumiranim podacima u toku istog dana i prosečnom brzinom, i ispiše rezultate. Na kraju, program dealocira svu korišćenu dinamičku memoriju.

Primeri

Ulaz	Izlaz
10, 0.98, 30/04 40, 4, 30/04 55, 7, 05/05 20, 2.98, 10/05 31, 3.21, 10/05	50, 4.98, 30/04, 5.98 55, 7, 05/05, 7.64 51, 6.19, 10/05, 7.28
31, 3.21, 10/05 55, 7, 05/05 10, 0.98, 30/04 20, 2.98, 10/05 40, 4, 30/04	50, 4.98, 30/04, 5.98 55, 7, 05/05, 7.64 51, 6.19, 10/05, 7.28
31, 3.21, 01/01 55, 7, 05/05 20, 2.98, 01/01 40, 4, 05/05	51, 6.19, 01/01, 7.28 95, 11, 05/05, 6.95