

## 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:

- a) učitava sve potrebne podatke;
- b) izvrši zahtevanu obradu;
- c) 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:

1. Rok za predaju četvrtog domaćeg zadatka je **ponedeljak, 20.05.2024.** 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.
2. Domaći zadaci će biti testirani i ocenjivani korišćenjem javnih i tajnih testova.
3. 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.
4. 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.
5. Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preduzmu odgovarajuće disciplinske mere.
6. Svi drugi detalji oko predaje, ocenjivanja i odbrane 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):  **$i = (R + G) \bmod 4$**
8. Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:
  - **dz4.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

13.05.2024. godine

*sa predmeta*

0. Napisati program na programskom jeziku C koji agregira logove pristupa veb severu. Program treba da učitava **id** korisnika koji pristupa veb serveru u jednom redu (pozitivan ceo broj), a zatim i sve logove tog korisnika u zasebnim redovima. Jedan log se sastoji iz vremena pristupa u formatu **HH:MM** (gde **HH** predstavlja sate, a **MM** minute pristupa sajtu) i adresi stranice kojoj je korisnik pristupio. Kraj unosa logova se označava sa **###**. Logovi jednog korisnika su sortirani po vremenu pristupa veb serveru. Program treba da učitava logove korisnika i vrši agregaciju sa već postojećim logovima sve dok se za **id** korisnika ne unese vrednost 0. Agregacija logova se vrši tako da se svi logovi čuvaju sortirano po vremenu pristupa, odnosno po **id** korisnika u slučaju da su 2 ili više korisnika pristupila veb serveru u isto vreme. Ukoliko je vreme pristupa veb serveru van opsega potrebno je ispisati poruku **ERROR** i korektno završiti program.

Program napisati prema sledećim stavkama.

- 1) Implementirati funkciju koja učitava logove jednog korisnika i formira jednostruko ulančanu listu logova. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja agregira logove jednog korisnika sa prethodno sačuvanim logovima po navedenom kriterijumu.
- 3) Implementirati funkciju koja ispisuje podatke iz ulančane liste na standardni izlaz, svaki u zasebnom redu. Nakon liste, funkcija ispisuje **###** u jednom redu.
- 4) Implementirati funkciju koja oslobađa zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija čita podatke o logovima korisnika, agregira ih sa postojećim logovima i ispisuje novu listu logova na standardni izlaz. Obradu ponavljati sve dok se za **id** korisnika ne unese vrednost 0.

## Primeri

Ulaz:	Izlaz:
123 12:00 https://rti.etf.bg.ac.rs/ 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 0	123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ###
123 12:00 https://rti.etf.bg.ac.rs/ 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 777 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 0	123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 777 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ ###
123 12:00 https://rti.etf.bg.ac.rs/ 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 56:73 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 0	ERROR

1. Napisati program na programskom jeziku C koji iz svih logova pristupa veb severu izdvaja one koji se odnose na pristup određenoj stranici. Program prvo treba da učitava logove o svim pristupima veb serveru. Jedan log se sastoji iz **id** korisnika koji je pristupio veb serveru (pozitivan ceo broj), vremena pristupa u formatu **HH:MM** (gde **HH** predstavlja sate, a **MM** minute pristupa sajtu) i adresi stranice kojoj je korisnik pristupio. Kraj unosa logova se označava sa **###**. Program zatim treba da učitava veb stranice, iz svih logova izdvoji pristupe toj stranici i ponavlja datu obradu do kraja unosa. Ukoliko je **id** korisnika koji pristupa veb serveru van opsega 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 logovima. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja iz liste logova izdvaja sve logove koji se odnose na određenu veb stranicu. Povratna vrednost je pokazivač na listu sa izdvojenim logovima, a ti logovi više nisu deo originalne liste.
- 3) Implementirati funkciju koja ispisuje podatke iz ulančane liste na standardni izlaz, svaki u zasebnom redu. Nakon liste, funkcija ispisuje **###** u jednom redu.
- 4) Implementirati funkciju koja oslobađa zauzetu dinamičku memoriju.
- 5) Napisati glavni program koji učitava listu logova, a zatim učitava imena veb stranice, ispisuje izdvojene logove na standardni izlaz i ponavlja obradu sve do kraja standardnog unosa. Nakon toga, glavni program ispisuje preostale logove.

### Primeri

Ulaz:	Izlaz:
777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### https://rti.etf.bg.ac.rs/rti/irlp1/ https://rti.etf.bg.ac.rs/rti/irlp2/	777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ ### 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 123 12:00 https://rti.etf.bg.ac.rs/ ###
777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### https://rti.etf.bg.ac.rs/rti/irlp4/ https://rti.etf.bg.ac.rs/rti/irlp2/	### 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ ###
777 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ -91 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ ###	ERROR

2. Napisati program na programskom jeziku C koji na osnovu svih logove pristupa veb severu pronalazi veb stranicu kojoj je pristupljeno najviše puta. Program prvo treba da učita logove o svim pristupima veb serveru. Jedan log se sastoji iz **id** korisnika koji je pristupio veb serveru (pozitivan ceo broj), vremena pristupa u formatu **HH:MM** (gde **HH** predstavlja sate, a **MM** minute pristupa sajtu) i adresi stranice kojoj je korisnik pristupio. Logovi se učitavaju do kraja unosa. Logovi su sortirani po adresi stranice, odnosno po **id** korisnika ukoliko su adrese jednake. Program zatim poziva funkciju koja sve elemente liste koji se odnose na istu veb stranicu spaja u jedan element koji dodatno sadrži broj pristupa toj stranici. Na kraju, program pronalazi i ispisuje veb stranicu kojoj je pristupljeno najviše puta. Ukoliko postoji veći broj stranica kojima je pristupljeno najviše puta ispisati sve u zasebnim redovima. Ukoliko je vreme pristupa veb serveru van opsega 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 logovima. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja spaja elemente liste koji se odnose na istu veb stranicu.
- 3) Implementirati funkciju koja ispisuje podatke iz ulančane liste na standardni izlaz, svaki u zasebnom redu. Nakon liste, funkcija ispisuje **###** u jednom redu.
- 4) Implementirati funkciju koja ispisuje sve veb stranice sa najveći brojem pristupa.
- 5) Implementirati funkciju koja oslobađa zauzetu dinamičku memoriju.
- 6) Napisati glavni program koji učitava listu logova, pozove funkciju za spajanje elemenata liste, ispiše novu listu i pozove funkciju za ispis imena stranica sa najvećim brojem pristupa.

### Primeri

<b>Ulaz:</b>	<b>Izlaz:</b>
123 12:00 https://rti.etf.bg.ac.rs/ 77 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 223 12:10 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/	https://rti.etf.bg.ac.rs/ - 1 https://rti.etf.bg.ac.rs/rti/irlp1/ - 3 https://rti.etf.bg.ac.rs/rti/irlp2/ - 2 ### https://rti.etf.bg.ac.rs/rti/irlp1/
123 12:00 https://rti.etf.bg.ac.rs/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 223 12:10 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/	https://rti.etf.bg.ac.rs/ - 1 https://rti.etf.bg.ac.rs/rti/irlp1/ - 2 https://rti.etf.bg.ac.rs/rti/irlp2/ - 2 ### https://rti.etf.bg.ac.rs/rti/irlp1/ https://rti.etf.bg.ac.rs/rti/irlp2/
123 12:00 https://rti.etf.bg.ac.rs/ 77 56:97 https://rti.etf.bg.ac.rs/rti/irlp1/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/	ERROR

3. Napisati program na programskom jeziku C koji na osnovu svih logova pristupa veb severu ispisuje stranicu kojoj je zadati korisnik najranije pristupio. Program prvo treba da učitava logove o svim pristupima veb serveru. Jedan log se sastoji iz **id** korisnika koji je pristupio veb serveru (pozitivan ceo broj), vremena pristupa u formatu **HH:MM** (gde **HH** predstavlja sate, a **MM** minute pristupa sajtu) i adresi stranice kojoj je korisnik pristupio. Kraj unosa logova se označava sa **###**. Nakon toga program sortira sve logove najpre po vremenu pristupa, zatim po imenu stranice i na kraju po **id** korisnika. Program zatim treba da učitava **id** korisnika, u listi logova pronade i ispiše prvu stranicu kojoj je korisnik pristupio i ponavlja datu obradu do kraja unosa. Ukoliko je **id** korisnika koji pristupa veb serveru van opsega 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 logovima. Funkcija treba da elemente ulančava u listu u poretku učitavanja.
- 2) Implementirati funkciju koja sortira ulančanu listu po navedenom kriterijumu.
- 3) Implementirati funkciju koja ispisuje podatke iz ulančane liste na standardni izlaz, svaki u zasebnom redu. Nakon liste, funkcija ispisuje **###** u jednom redu.
- 4) Implementirati funkciju koja u sortiranoj listi pronalazi adresu veb stranice kojoj je zadati korisnik prvo pristupio.
- 5) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija učitava podatke sa standardnog ulaza, ispiše podatke na standardni izlaz, sortira podatke, ispiše podatke na standardni izlaz, a zatim učitava imena korisnika i ispisuje adrese stranica kojima su prvo pristupili.

### Primeri

Ulaz:	Izlaz:
123 12:00 https://rti.etf.bg.ac.rs/ 77 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:01 https://rti.etf.bg.ac.rs/rti/irlp1/ 223 12:10 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 12:02 https://rti.etf.bg.ac.rs/rti/irlp2/ 777 13:00 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 123	12:00 123 https://rti.etf.bg.ac.rs/ 10:00 77 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:01 123 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:10 223 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:02 123 https://rti.etf.bg.ac.rs/rti/irlp2/ 13:00 777 https://rti.etf.bg.ac.rs/rti/irlp2/ ### 10:00 77 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:00 123 https://rti.etf.bg.ac.rs/ 12:01 123 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:02 123 https://rti.etf.bg.ac.rs/rti/irlp2/ 12:10 223 https://rti.etf.bg.ac.rs/rti/irlp1/ 13:00 777 https://rti.etf.bg.ac.rs/rti/irlp2/ ### https://rti.etf.bg.ac.rs/
223 12:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 123 10:00 https://rti.etf.bg.ac.rs/ 123 12:00 https://rti.etf.bg.ac.rs/rti/irlp1/ 77 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ ### 123 77	12:00 223 https://rti.etf.bg.ac.rs/rti/irlp1/ 10:00 123 https://rti.etf.bg.ac.rs/ 12:00 123 https://rti.etf.bg.ac.rs/rti/irlp1/ 10:00 77 https://rti.etf.bg.ac.rs/rti/irlp1/ ### 10:00 123 https://rti.etf.bg.ac.rs/ 10:00 77 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:00 123 https://rti.etf.bg.ac.rs/rti/irlp1/ 12:00 223 https://rti.etf.bg.ac.rs/rti/irlp1/ ### https://rti.etf.bg.ac.rs/ https://rti.etf.bg.ac.rs/rti/irlp1/
-123 12:00 https://rti.etf.bg.ac.rs/ 77 10:00 https://rti.etf.bg.ac.rs/rti/irlp1/ ###	ERROR