

PRAKTIKUM IZ PROGRAMIRANJA 2

- domaći zadatak broj 3 -

Sastaviti program na programskom jeziku C koji vrši određenu obradu nad jednim ili više **nizova znakova (u daljem tekstu stringova)**. 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.

Prilikom ispisa na kraj svakog stringa dodati znak za novi red. Dimenzije pojedinačnih stringova nisu poznate unapred. Memorija predviđena za string mora biti dinamički alocirana. Količina memorije koja je odvojena za string mora biti takva da iskorišćenje na kraju učitavanja bude potpuno (ni bajt više, ni bajt manje). Realokaciju vršiti uvek kada je potrebno, i prilikom unosa stringa i prilikom obrade. U toku rada, prilikom svake alokacije ili realokacije dinamičke memorije, proveravati uspešnost poziva alloc funkcije. U slučaju neuspešne dodele dinamičke memorije, u glavnom programu ispisati poruku MEM_GRESKA i korektno prekinuti izvršavanje (vraćanjem vrednosti 0 kao rezultata izvršavanja programa).

Nakon obrade (unos, ispis unetih podataka, obrada, ispis dobijenih rezultata) dealocirati svu dinamičku memoriju. Smatrati da korisnik pojedinačan string zadaje u jednom redu teksta putem standardnog ulaza (u string ulaze svi znakovi do $\backslash n$, ne uključujući i $\backslash n$). Obradu treba vršiti nad originalnim stringom (ne nad kopijom), osim u slučaju formiranja novog stringa ili niza stringova, tamo gde je traženo.

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:

1. Rok za predaju drugog domaćeg zadatka je **ponedeljak, 15.05.2023.** 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.
2. Studentima je dostupno okruženje za testiranje rešenja domaćeg zadatka na Moodle platformi za elektronsko učenje korišćenjem javnih testova.
3. Prilikom predaje domaćeg zadatka studenti će rešavati i kratak test znanja u vezi sa rešenjem domaćeg zadatka i relevantnim gradivom iz programskog jezika C koje obuhvata temu domaćeg zadatka.
4. Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preduzmu odgovarajuće disciplinske mere.
5. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa): **$i = (R + G) \bmod 4$**
6. Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:
 - **dz3.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

0. Napisati program koji u nizu stringova konvertuje merne jedinice za merenje frekvencije (**Hz**), dužine (**m**), težine (**g**) i bajtova (**B**). Moguće je konvertovanje u sledeće prefikse za mernu jedinicu: mili (**m**, 10^{-3}), kilo (**k**, 10^3), mega (**M**, 10^6) i giga (**G**, 10^9). Pre stringova se unosi odgovarajuća vrednost stepena (-3, 3, 6 ili 9) u koji je potrebno izvršiti konverziju. Ukoliko se unese neadekvatan broj ispisati poruku **GRESKA**. U stringovima se odmah nakon celobrojne ili realne vrednosti nalazi prefiks (ako postoji), a zatim i sama merna jedinica. Pretpostaviti da će nakon konverzije rezultat uvek biti ceo broj. Za konverziju stringa u realan broj može se koristiti funkcija `atof()` koja se koristi na isti način kao i funkcija `atoi()`. Broj stringova, odnosno redova, nije unapred poznat.

Program napisati prema sledećim stavkama. Prilikom izrade pojedinačne stavke pretpostaviti da su funkcije iz stavki koje su prethodile trenutnoj date i da njihove funkcionalnosti odgovaraju datom opisu.

- 1) Implementirati funkciju `char* readLine()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** readLines(int *n)`; koja čita stringove. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void updateLines(int convertTo, char **lines, int n)`; koji u nizu stringova `lines` dužine `n` ažurira sve vrednosti prefiks zadat preko `convertTo` prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita vrednost prefiksa i stringove sa standardnog ulaza, menja sve vrednosti mernog sistema na opisani način i ispiše rezultat na standardni izlaz.

Primeri

Ulaz:	Izlaz:
-3 Intel Core i5-10500 3.10GHz (do 4.50GHz) HP 9mm Slim DVD-Writer 0.155m x 0.337m x 0.303m 4.7kg	Intel Core i5-10500 3100000000000mHz (do 4500000000000mHz) HP 9mm Slim DVD-Writer 155mm x 337mm x 303mm 4700000mg
3 Intel Core i7-12700K 2.70GHz (do 5.00GHz) 16GB (1 x 16GB) DDR5 4800MHz RAM, 4 DIMM slota 512GB PCIe M.2 SSD 7000g	Intel Core i7-12700K 2700000kHz (do 5000000kHz) 16000000kB (1 x 16000000kB) DDR5 4800000kHz RAM, 4 DIMM slota 512000000kB PCIe M.2 SSD 7kg
-1	GRESKA

1. Napisati program koji u nizu stringova konvertuje obim pizze u njenu površinu. Pretpostaviti da je pizza oblika savšenog kruga. Izračunatu površinu zaokružiti na prvi manji ceo broj. Pre stringova se unosi realna vrednost broja pi. Ukoliko se unese realna vrednost čiji ceo deo je različit od 3 ispisati poruku **GRESKA**. U stringovima se odmah nakon obima pizze nalaze karakteri **cm**. Broj stringova, odnosno redova, nije unapred poznat.

Program napisati prema sledećim stavkama. Prilikom izrade pojedinačne stavke pretpostaviti da su funkcije iz stavki koje su prethodile trenutnoj date i da njihove funkcionalnosti odgovaraju datom opisu.

- 1) Implementirati funkciju `char* readLine()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** readLines(int *n)`; koja čita stringove. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void toArea(float pi, char **lines, int n)`; koja u nizu stringova `lines` dužine `n` menja sve obime u površinu prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita realnu vrednost pi i stringove sa standardnog ulaza, menja sve obime u površine na opisani način i ispiše rezultat na standardni izlaz.

Primeri

Ulaz:	Izlaz:
3.14 Srednja pizza je 27cm. Malo veca je velika pizza od 32cm. Zatim sledi extra velika od 42cm. Najveca pizza je 50cm.	Srednja pizza je 58cm ² . Malo veca je velika pizza od 81cm ² . Zatim sledi extra velika od 140cm ² . Najveca pizza je 199cm ² .
3 Standardna pizza od 30cm kosta oko 1000 dinara. Porodicna pizza koja je 40cm kosta oko 500 dinara vise.	Standardna pizza od 75cm ² kosta oko 1000 dinara. Porodicna pizza koja je 133cm ² kosta oko 500 dinara vise.
4	GRESKA

2. Napisati program koji u nizu stringova konvertuje dijagonalu ekrana u njegovu širinu i visinu. Pretpostaviti da je ekran oblika pravougaonika. Izračunate dimenzije zaokružiti na prvi manji ceo broj. Pre stringova se unosi odnos širine i visine ekrana u formatu $\text{š}:\text{v}$ gde su š i v celi brojevi. Ukoliko nije ispoštovan format ispisati poruku **GRESKA**. U stringovima se odmah nakon dužine dijagonale nalaze karakteri ' '. Za konverziju stringa u realan broj može se koristiti funkcija `atof()` koja se koristi na isti način kao i funkcija `atoi()`. Broj stringova, odnosno redova, nije unapred poznat.

Program napisati prema sledećim stavkama. Prilikom izrade pojedinačne stavke pretpostaviti da su funkcije iz stavki koje su prethodile trenutnoj date i da njihove funkcionalnosti odgovaraju datom opisu.

- 1) Implementirati funkciju `char* readLine()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** readLines(int *n)`; koja čita stringove. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void toWidthHeight(int width, int height, char **lines, int n)`; koja u nizu stringova `lines` dužine `n` menja dijagonale u širinu i visinu ekrana prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita odnos širine i visine i stringove sa standardnog ulaza, menja sve dijagonale u širinu i visinu na opisani način i ispiše rezultat na standardni izlaz.

Primeri

Ulaz:	Izlaz:
4:3 LG OLED65C21LA Smart televizor je dostupan u dijagonalama od 42'', 65'' i 77''. Dijagonala od 55'' je jedina dostupna za televizor LG OLED55A23LA Smart televizor.	LG OLED65C21LA Smart televizor je dostupan u dijagonalama od 33x25, 52x39 i 61x46. Dijagonala od 44x33 je jedina dostupna za televizor LG OLED55A23LA Smart televizor.
16:9 Laptop HP 15s-eq3008nm - 65A92EA ima dijagonalu od 15.6''.	Laptop HP 15s-eq3008nm - 65A92EA ima dijagonalu od 13x7.
21-9	GRESKA

3. Napisati program koji u nizu stringova konvertuje vreme u traženi format. Vreme može biti zadato u danima (**d**an), satima (**h**), minutima (**min**) i sekundama (**s**). Pre stringova se unosi informacija u čemu je zadato vreme u stringovima. Svaki ceo broj u stringovima treba tumačiti kao vreme i konvertovati u format **d**an:**h**:**min**:**s**. Ukoliko se unese neadekvatna oznaka ispisati poruku **GRESKA**. Broj stringova, odnosno redova, nije unapred poznat.

Program napisati prema sledećim stavkama. Prilikom izrade pojedinačne stavke pretpostaviti da su funkcije iz stavki koje su prethodile trenutnoj date i da njihove funkcionalnosti odgovaraju datom opisu.

- 1) Implementirati funkciju `char* readLine()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** readLines(int *n)`; koja čita stringove. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void formatTime(char *givenTime, char **lines, int n)`; koja u nizu stringova `lines` dužine `n` menja sve cele brojeve u vremenski format prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita informaciju o zadatom vremenu i stringove sa standardnog ulaza, menja sve cele brojeve u traženi format na opisani način i ispiše rezultat na standardni izlaz.

Primeri

Ulaz:	Izlaz:
min Ispit traje 180. Kolokvijum je trajao 90.	Ispit traje 0:3:0:0. Kolokvijum je trajao 0:1:30:0.
s Eliud Kipchoge (Kenya) 7269; Kelvin Kiptum (Kenya) 7285; Kenenisa Bekele (Ethiopia) 7301;	Eliud Kipchoge (Kenya) 0:2:1:9; Kelvin Kiptum (Kenya) 0:2:1:25; Kenenisa Bekele (Ethiopia) 0:2:1:41;
ms	GRESKA