

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:

- 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.

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 `malloc` 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 `\n`, ne uključujući i `\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:

- Rok za predaju trećeg domaćeg zadatka je **utorak, 18.05.2021.** 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:
 - dz3.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

0. Napisati program koji u nizu stringova koji sadrže kôd na nekom programskom jeziku uklanja sve komentare. Komentar se uvek nalazi između oznaka za početak i kraj komentara koje korisnik unosi sa standardnog ulaza i koje se mogu sastojati od više znakova, ali ne mogu biti prazan string. Na primer, to su oznake `/*` i `*/` na programskom jeziku C, oznake `{` i `}` na programskom jeziku Pascal ili oznake `'` i `'` na programskom jeziku Python. Program najpre učitava oznake za početak i kraj komentara, svaki string u zasebnom redu. Zatim se unose stringovi koji predstavljaju kod sa komentarima. Komentar može da se proteže na više redova. Ukoliko se ne unesu oznake početka i kraja komentara i bar jedan string sa programskom kodom ispisati poruku **GRESKA**. Broj stringova, odnosno redova, nije unapred poznat. Kraj unosa se označava praznim redom.

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 koji sadrže programski kod sa komentarima. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void removeComments(char *beginComment, char *endComment, char **code, int n)`; koji iz niza stringova `code` dužine `n` uklanja sve komentare.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita oznake početka i kraja komentara i stringove sa kodom sa standardnog ulaza, uklanja sve komentare na opisani način i ispiše rezultat na standardni izlaz. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
<pre>/* */ #include <stdio.h> /* Glavni program */ int main() { printf("Hello World"); /* Ispis*/ return 0; }</pre>	<pre>#include <stdio.h> int main() { printf("Hello World"); return 0; }</pre>
<pre>{ } {Welcome to GDB Online.} program Hello; begin writeln ('Hello World'); {readln} end.</pre>	<pre>program Hello; begin writeln ('Hello World'); end.</pre>
<pre>/* */</pre>	<pre>GRESKA</pre>

1. Napisati program koji vrši sakrivanje mobilnih telefonskih brojeva na osnovu zadatog niza stringova koji može sadržati i druge znake. Mobilni telefonski broj obavezno počinje prefiksom 06, nakon čega slede brojevi koji mogu biti razdvojeni razmakom, crticom – ili kosom crtom /. Sakrivanje broja se vrši tako što se najpre uklone eventualni razmaci, crtice ili kose crte, a zatim poslednje tri cifre zamene stringom za zamenu koji se unosi sa standardnog ulaza. Ostali brojevi telefona treba da ostanu neizmenjeni. Program najpre treba da učitava string za zamenu koji može biti prazan. Zatim se unose stringovi koji mogu sadržati telefonske brojeve. Smatrati da se broj nalazi u jednom redu. Kraj unosa se označava praznim redom. U slučaju da je niz stringova koji se obrađuje prazan ispisati poruku **GRESKA**.

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 niz stringova koji mogu sadržati telefonske brojeve. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `char** changeNumber(char *replace, char **strings, int n)`; koja formira novi niz stringova prema opisanoj obradi na osnovu stringa za zamenu `replace` i niza stringova `strings`. Funkcija putem povratne vrednosti vraća novoformirani niz.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita string za zamenu i niz stringova koji mogu sadržati mobilne telefonske brojeve sa standardnog ulaza, a zatim formira i ispiše novi niz stringova u kome će svi mobilni telefonski brojevi biti skriveni na opisani način. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
*** Pozovi Petrovica na 064 555 333 064/211-111 Mikin broj	Pozovi Petrovica na 064555*** 064211*** Mikin broj
scramble 063/111/5556 0 1 1 2 2 2 3 3 3	0631115scramble 0 1 1 2 2 2 3 3 3
scramble	GRESKA

2. Napisati program koji konverziju novčanih iznosa u stringu sa porukom iz jedne valute u drugu. Novčani iznos je realan broj nakon koga neposredno sledi jedan blanko znak i troslovna oznaka valute. Program najpre učitava string u kome se vrši zamena. Zatim učitava string koji predstavlja troslovnu oznaku valute u koju se radi konverzija. Nakon toga učitava niz stringova koji predstavlja kursnu listu, gde svaki sadrži troslovnu oznake valute iz koje se konvertuje i vrednost konverzije u valutu u koju se konvertuje koja mora predstavljati realan broj. U slučaju da je string koji se obrađuje prazan ili valuta iz poruke ne postoji u kursnoj listi ispisati poruku **GRESKA**. Zaokruživanja prilikom konverzije valuta vršiti na dve decimale.

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()`; koja čita kursnu listu. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Kraj niza pokazivača označava se `NULL` pokazivačem.
- 3) Implementirati funkciju `char* convertMoney(char *msg, char *currencyCode char **exchgValues)`; koja formira i vraća novi string na osnovu stringa `msg` tako što sve pojave novčanih iznosa u valutama zadatim u nizu stringova `exchgValues` menja odgovarajućim novčanim iznosom koji prati oznaka valute `currencyCode`.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita string sa porukom, oznaku valute i kursnu listu sa standardnog ulaza, a zatim formira i ispiše novu poruku za izmenjenim novčanim iznosima u traženoj valuti. Nakon svake pojave staviti znak za novi red.

Primeri

Ulaz:	Izlaz:
Cena najma apartmana je 10.0 EUR tokom radne nedelje RSD EUR 118.1 GBP 137.07	Cena najma apartmana je 1181.00 RSD tokom radne nedelje
Coffee 2.0 EUR cold drinks 3.0 USD RSD EUR 118.1 GBP 137.07 USD 97.0	Coffee 236.20 RSD cold drinks 291.00 RSD
RSD EUR 118.1	GRESKA

3. Napisati program koji u zadatom nizu stringova pronalazi i ispisuje sve parove stringova koji se sastoje od istih slova engleskog alfabeta. Prilikom obrade zanemariti razliku između malih i velikih slova, kao i prisustvo drugih znakova. Broj stringova nije unapred poznat. Kraj unosa se označava praznim redom. Ukoliko je broj unetih stringova 0 ispisati poruku **GRESKA**.

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()`; koja vraća pokazivač na niz stringova pročitanih sa standardnog ulaza. Kraj niza pokazivača označava se `NULL` pokazivačem.
- 3) Implementirati funkciju `int areSameLetters(char *string0, char *string1)`; koja proverava da li se stringovi `string0` i `string1` sastoje od istih slova na opisani način. Ukoliko predstavljaju funkcija vraća 1, a u suprotnom 0.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita niz stringova i ispiše sve parove stringova koji se sastoje od istih slova prema zadatom kriterijumu. Na kraju svakog reda dodati znak za novi red, a same stringove razdvojiti znakom '|'. .

Primeri

Ulaz:	Izlaz:
Evil* vi-le a-gentleman elegant MAN pat LIVE	Evil* vi-le Evil* LIVE vi-le LIVE a-gentleman elegant MAN
šbrakadšbra barakuda%% bara dabar u krdu burad abrak*dabra	šbrakadšbra barakuda%% šbrakadšbra dabar u krdu barakuda%% dabar u krdu
	GRESKA

4. Napisati program koji vrši ubacivanje emotikona („smajlija“) umesto odgovarajuće oznake koja se nalazi u zadatom stringu sa porukom. Oznake emotikona obavezno počinju znakom dvotačka, nakon čega sledi reč kojom se on zadaje bez blanko znakova. Program najpre učitava string u kome se vrši ubacivanje emotikona. Nakon toga se učitava niz stringova čiji svaki string sadrži parove reči koje predstavljaju oznaku emotikona i odgovarajuću sekvencu kojom se on menja. Ukoliko oznaka emotikona postoji u stringu sa porukom, a ne postoji u nizu stringova za zamenu, zamenu izvršiti sa sekvencom : | . Broj stringova, odnosno redova, nije unapred poznat. Kraj unosa se označava praznim redom. Ukoliko je broj unetih stringova 0 ispisati poruku **GRESKA**.

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 koji sadrže parove oznaka emotikona i sekvence za zamenu. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `char* changeEmoticons(char *string, char **values, int n)`; koja vrši zamenu oznaka emotikona u zadatom stringu sa porukom `msg` tako što svaku pojavu reči koja predstavlja emotikon menja odgovarajućom sekvencom za zamenu iz niza stringova `values` dužine `n`.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita string sa porukom i parove stringove za zamenu sa standardnog ulaza, vrši zamenu na opisani način i ispiše rezultat na standardni izlaz. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
<pre>Bas si :srce lutka :smesko :smesko :-) :ljutko ;(:tugica :-(:mangup ;) :srce <3</pre>	<pre>Bas si <3 lutka :-)</pre>
<pre>:smesko :smesko :andjeo :smesko :-) :ljutko ;(</pre>	<pre>:-) :-) : </pre>
	GRESKA