

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 `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:

1. Rok za predaju trećeg domaćeg zadatka je **utorak, 17.05.2022.** 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): **$i = (R + G) \bmod 5$**
8. 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 sadrži datume po formatu `d-m-g`, gde `d` označava dan, `m` označava mesec, a `g` godinu koji ne moraju sadržati vodeće nule, vrši ažuriranje svih datuma za zadati broj dana unapred i formatira datume uz korišćenje vodećih nula po formatu `dd-mm-gggg`. Godina se smatra prestupnom ukoliko je broj godine deljiv sa četiri i nije deljiv sa 100 ili broj godine je deljiv sa 400. Program najpre učitava broj dana za koji se datumi pomeraju unapred. Zatim se unose stringovi koji mogu sadržati datume u navedenom obliku, kao i drugi tekst. Ukoliko se ne unese broj dana za ažuriranje u obliku nenegativnog celog broja i bar jedan string sa tekstom 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 mogu sadržati datume. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void updateDates (int dayOffset, char **stringDates, int n)`; koji u nizu stringova `stringDates` dužine `n` ažurira sve datume za `dayOffset` unapred prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita broj dana za pomeranje i stringove sa standardnog ulaza, menja sve datume na opisani način i ispiše rezultat na standardni izlaz. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
1 Today is 31-12-2022 Tomorrow, 1-1-2023 is New Years's day.	Today is 01-01-2023 Tomorrow, 02-01-2023 is New Years's day.
17 17-05-2022 homework submission Defense 18-05-2022 in lab 70 PP2 exam 14-06-2022 P2 exam 15-06- 2022 June exam period ends on 19-06.	03-06-2022 homework submission Defense 04-06-2022 in lab 70 PP2 exam 01-07-2022 P2 exam 02-07- 2022 June exam period ends on 19-06.
-5	GRESKA

1. Napisati program koji u nizu stringova koji sadrži datume po formatu `ddSmmSgggg` u format `Month dth, gggg`, tako da umesto `mm` (što označava mesec) stoji naziv meseca (`04` se menja sa `April`), a `dd` (što označava redni broj dana u mesecu) se zameni rednim brojem dana uz dodavanje odgovarajućeg sufiksa rednog broja (`st, nd, rd` ili `th`). Znak `s` predstavlja separator datuma i može biti neki od znakova `./-`, što se unosi sa standardnog ulaza. Program najpre učitava znak koji predstavlja separator datuma. Zatim se unose stringovi koji mogu sadržati datume u navedenom obliku, kao i drugi tekst. Ukoliko se ne unese separator datuma i bar jedan string sa tekstom 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 mogu sadržati datume. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `void changeDates (char separator, char **string_dates, int n)`; koji u nizu stringova `string_dates` dužine `n` menja sve datume prema navedenom pravilu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita separator datuma i stringove sa standardnog ulaza, menja sve datume na opisani način i ispiše rezultat na standardni izlaz. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
. Today is 06.05.2022	Today is May 6th, 2022
- 17-05-2022 homework submission Defense 18-05-2022 in lab 70 PP2 exam 14-06-2022 P2 exam 15-06-2022 June exam period ends on 19-06.	May 17th, 2022 homework submission Defense May 18th, 2022 in lab 70 PP2 exam June 14th, 2022 P2 exam June 15th, 2022 June exam period ends on 19-06.
%	GRESKA

2. Napisati program koji vrši sakrivanje adresa elektronske pošte na osnovu zadatog niza stringova koji može sadržati i druge znake. Smatrati da je adresa elektronske pošte isključivo u obliku `username@domain`, gde `username` i `domain` mogu sadržati isključivo mala slova engleske abecede i tačku. Sakrivanje adrese elektronske pošte se vrši tako što se adresa zameni stringom `username[at]domain` uz navođenje odgovarajućeg broja blanko znakova nakon svakog karaktera u stringovima `username` i `domain` i stringa `[at]` koji se umeće umesto znaka `@`. Program najpre treba da učitava broj blanko znakova koji se ugrađuje u sakrivenu adresu. Zatim se unose stringovi koji mogu sadržati adrese elektronske pošte. Smatrati da se adresa uvek nalazi u jednom redu. Kraj unosa se označava praznim redom. U slučaju da je broj blanko znakova negativan ili 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 adrese elektronske pošte. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `char** obfuscateEmail(int blanks, char **strings, int n)`; koja formira novi niz stringova prema opisanoj obradi na osnovu broja blanko znakova `blanks` i niza stringova `strings`. Funkcija putem povratne vrednosti vraća novoformirani niz.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita broj blanko znakova i niz stringova koji mogu sadržati adrese elektronske pošte sa standardnog ulaza, a zatim formira i ispiše novi niz stringova u kome će sve adrese elektronske pošte biti skrivene na opisani način. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
0 Adresa mika@yahoo.com je blokirana Javite se na admin@io.port	Adresa mika[at]yahoo.com je blokirana Javite se na admin[at]io.port
1 mika@yahoo.com pera@etf.rs Portir nesa@rcp.etf.bg.ac.rs	m i k a [at] y a h o o . c o m p e r a [at] e t f . r s P o r t i r n e s a [at] r c p . e t f . b g . a c . r s
-2	GRESKA

3. Napisati program koji u zadatim stringovima vrši konverziju `for` petlje na programskom jeziku C u `while` petlju. Petlja `for` ima standardan oblik `for(preduslov;uslov;postuslov) {...}` uz moguće postojanje dodatnih blanko znaka, gde `...` predstavlja bilo kakav sadržaj koji čini telo petlje. Program treba da na osnovu petlje navedenog oblika formira ekvivalentnu `while` petlju uz uklanjanje svih blanko znakova, kako u okviru definicije, tako i u okviru tela petlje. Smatrati da se petlje ne ugnežavaju, a da string može sadržati i drugi tekst koji predstavlja ispravan programski kod na programskom jeziku C. Program učitava stringove koji sadrže programski kod na programskom jeziku C. Smatrati da se `for` petlja uvek 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()`; koja čita niz stringova koji sadrže programski kod na programskom jeziku C. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Kraj niza pokazivača označava se `NULL` pokazivačem.
- 3) Implementirati funkciju `char** forToWhile(char **srcCode)`; koja formira novi niz stringova prema opisanoj obradi na osnovu niza stringova `srcCode`. Funkcija putem povratne vrednosti vraća novoformirani niz.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita niz stringova koji sadrže programski kod na programskom jeziku C, a zatim formira i ispiše novi niz stringova u kome će sve pojave `for` petlje biti zamenjene `while` petljom na opisani način. Nakon ispisa dodati znak za novi red.

Primeri

Ulaz:	Izlaz:
<pre>for(i=0;i<n;i++) { a[i]+=b[i]; }</pre>	<pre>i=0;while(i<n){a[i]+=b[i];i++;}</pre>
<pre>sum=10; a=5; for(d=c=0; c < 10 ; c ++) { sum+=a;} print("%d", sum);</pre>	<pre>sum=10; a=5; d=c=0;while(c<10){sum+=a;c++;} print("%d", sum);</pre>
	GRESKA

4. Napisati program koji u zadatim stringovima vrši konverziju vremena u obliku `hh:mm TZ`, gde `hh` predstavlja sate, `mm` minute, a `TZ` oznaku vremenske zone (dužine 2-4 znaka) u zadatu vremensku zonu. Program najpre učitava string u kome se vrši zamena. Zatim učitava string koji predstavlja vremensku zonu u koju se vrši konverzija. Nakon toga učitava niz stringova koji predstavljaju vremenske zone i njihov pomeraj u odnosu na koordinisano univerzalno vreme (UTC) koje efektivno predstavlja srednje vreme po Griniču. U slučaju da je string koji se obrađuje prazan ili vremenska zona iz poruke ne postoji u listi vremenskih zona 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.

- 5) Implementirati funkciju `char* readLine()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 6) Implementirati funkciju `char** readLines()`; koja čita listu vremenskih zona i njihovih pomeraja u odnosu na UTC. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Kraj niza pokazivača označava se `NULL` pokazivačem.
- 7) Implementirati funkciju `char* convertTime(char *msg, char *targetTimeZone char **timeZones)`; koja formira i vraća novi string na osnovu stringa `msg` tako što sve pojave vremenskih trenutaka zadatim u nizu stringova `timeZones` menja odgovarajućim vremenskim trenucima iz zone `targetTimeZone`.
- 8) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita string sa porukom, oznaku ciljane vremenske zone i listu vremenskih zona i njihovih pomeraja u odnosu na UTC sa standardnog ulaza, a zatim formira i ispiše novu poruku sa izmenjenim vremenima. Nakon svake pojave staviti znak za novi red.

Primeri

Ulaz:	Izlaz:
The meeting will start at 11:00 EST CET EST UTC-05 CET UTC+01	The meeting will start at 17:00 CET
The plane will depart at 11:00 KALT in the morning. First stop Prague 13:00 CET Second stop 15:00 AZOT in the afternoon. CET AZOT UTC-01 CET UTC+01 KALT UTC+02	The plane will depart at 10:00 CET in the morning. First stop Prague 13:00 CET Second stop 17:00 CET in the afternoon.
CET EST UTC-05 CET UTC+01	GRESKA