

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 `\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 drugog domaćeg zadatka je **utorak, 14.05.2024.** 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. 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;

07.05.2024. godine

sa predmeta

0. Napisati program koji u nizu stringova koji sadrže kod 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 ne 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* read_line ()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** read_lines (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 remove_comments (char *begin_comment, char *end_comment, 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(/* Ispis*/"Hello World"); 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>	<p>GRESKA</p>

1. Napisati program koji rotira rečenicu za zadati broj reči udesno ili ulevo. Na standardnom ulazu se najpre unosi rečenica u jednom ili više redova. Smatrati da su reči razdvojene tačno jednim blanko znakom. Kraj unosa se označava praznim redom. U sledećem redu se unosi broj reči za koji je neophodno rotirati rečenicu. Ukoliko je broj pozitivan rečenica se rotira udesno, a u suprotnom se rotira ulevo. Ukoliko se za rečenicu unese prazan string ispisati **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* read_line ()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** read_lines (int *n)`; koja čita stringove koji sa standardnog ulaza. Funkcija kao povratnu vrednost vraća niz pokazivača na unete stringove. Takođe, funkcija preko argumenta `n` vraća broj unetih vrednosti.
- 3) Implementirati funkciju `char* rotate_sentence (char **sentence, int n, int rotate)` koja rotira rečenicu za `rotate` reči. Rečenica je data nizom stringova `sentence` dužine `n`. Funkcija kao rezultat vraća pokazivač na novoformiranu rečenicu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita rečenicu sa standardnog ulaza, rotira je za zadati broj reči i ispiše je na standardnom izlazu. Reči razdvojiti blanko znakom bez znaka za novi red nakon poslednje reči.

Primeri

Ulaz:	Izlaz:
to be or not to be 3	not to be to be or
to be or not to be -2	or not to be to be
4	GRESKA

2. Napisati program koji vrši kompresiju niza stringova. Kompresija stringa se vrši na osnovu frekvencije pojavljivanja karaktera. Svaka sekvenca uzastopnih pojavljivanja istog karaktera se menja tim karakterom i brojem pojavljivanja. Ukoliko nije unet nijedan string 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* read_line ()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** read_words ()`; koja korišćenjem prethodne funkcije čita reči sa standardnog ulaza. Funkcija kao povratnu vrednost vraća niz pokazivača na unete reči. Poslednji element niza treba da ima vrednost **NULL**.
- 3) Implementirati funkciju `char* compress (char *word)`; koja niz karaktera `word` kompresuje na opisani način. Funkcija vraća pokazivač na novoformiranu kompresovanu reč.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita reči sa standardnog ulaza, izvrši kompresiju svake i ispiše rezultat na standardni izlaz.

Primeri

Ulaz:	Izlaz:
wwwaaadexxxxxx	w4a3d1e1x6
Lorem Ipsum is simply dummy text	L1o1r1e1m1 I1p1s1u1m1 i1s1 s1i1m1p1l1l1y1 d1u1m2y1 t1e1x1t
	GRESKA

3. Napisati program koji konvertuje brojeve u dekadni brojevni sistem. Brojevi su zadati u više linije u formatu **vrednost (osnova) , vrednost (osnova) ...** Osnove mogu biti u opsegu od 2 do 20. Cifre čije vrednosti se vrednosti nalaze u opsegu od 10 do 19 predstavljene su slovima engleskog alfabeta u opsegu od A do K, respektivno. Dužina jedne linije i ukupan broj linija nisu poznati unapred. Kraj unosa se označava praznim redom. Ukoliko nije unet nijedan string 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* read_line ()`; koja čita jedan red sa standardnog ulaza i vraća pokazivač na taj učitani red.
- 2) Implementirati funkciju `char** read_numbers ()`; koja korišćenjem prethodne funkcije čita brojeve sa standardnog ulaza. Funkcija kao povratnu vrednost vraća niz pokazivača na unete brojeve. Poslednji element niza treba da ima vrednost `NULL`.
- 3) Implementirati funkciju `double convert (char *number, int base)`; koja broj `number` dat kao niz znakova u osnovi `base` konvertuje u realan broj u dekadnom brojevnom sistemu.
- 4) Napisati glavni program koji korišćenjem prethodno realizovanih funkcija pročita informaciju o brojevima sa standardnog standardnog ulaza, a potom konvertuje sve brojeve u dekadni brojevni sistem i ispiše rezultat na standardni izlaz. Brojeve zaokružiti na tri decimale.

Primeri

Ulaz:	Izlaz:
20 (20)	40.000
7 (7) , 1.A (16) 10 (5)	7.000 16.625 5.000
	GRESKA