

## PRAKTIKUM IZ PROGRAMIRANJA 2

### - domaći zadatak broj 3 -

Sastaviti program na programskom jeziku C kojim se vrši određena vrsta obrade nad znakovnim nizovima (u daljem tekstu, stringovi) koji se unose iz proizvoljnog broja redova sa standardnog ulaza. Program treba da repetitivno učitava ulazne stringove i pamti ih u dinamički niz stringova, dok se ne unese prazan red, a zatim učitava eventualne dodatne podatke prema potrebama zadatka. Po učitavanju, program treba da ispiše učitane stringove, izvrši zahtevanu obradu nad svakim stringom iz niza pojedinačno, ispiše rezultujuće stringove i nakon toga oslobodi zauzetu dinamičku memoriju. Navedene korake u programu ponavljati sve dok korisnik za neki od ulaznih stringova ne unese „dosta Brus Li”.

Zavisno od rednog broja problema, sastaviti **jedan** od sledećih programa koji:

0. menja sve indentifikatore u ulaznom stringu kod kojih je više reči u nazivu odvojeno znakom `_` tako što sve reči u nazivu spaja, a svaka reč započinje velikim slovom; npr. `counter_strike` menja sa `counterStrike`; identifikatori su razdvojeni sa jednim ili više blanko znakova;
1. vrši inverznu izmenu od one iz tačke 0; smatrati da svako veliko slovo u indentifikatoru osim početnog predstavlja početak nove reči; npr. `counterStrike` se menja u `counter_strike`, `checkFPS` u `check_f_p_s`;
2. uklanja iz stringa sve pojave rezervisanih reči jezika Pascal za rad sa petljama (`for`, `while`, `do`, `repeat`, `until`); prilikom uklanjanja, ignorisati razliku između velikih i malih slova;
3. svaki string koji predstavlja `for` petlju napisanu na Pascal-u menja for petljom na jeziku C; npr. `for i:=n downto 1 do` se menja sa `for(i=n; i>=1; i--)`; pretpostaviti da svaki string koji počinje ključnom reči `for` predstavlja korektnu `for` naredbu;
4. svaki string koji predstavlja `if` kontrolnu strukturu napisanu na Pascal-u menja odgovarajućim konstruktom na jeziku C; npr. `if i < 10 then begin <blok koda> end`; se menja sa `if (i < 10) { <blok koda> }`; pretpostaviti da svaki string koji počinje ključnom reči `if` predstavlja korektnu `if` naredbu i da `else` grana ne postoji;
5. u stringu koji sadrži kôd na programskom jeziku C menja sva indeksiranja oblika `*(p + i)` sa `p[i]`; konverziju ne treba vršiti ako se indeksiranje nalazi unutar komentara (`/*...*/`), a prilikom konverzije ignorisati blanko znake;
6. uklanja iz stringa sva poklapanja dela zadate reči za pretragu; reč za pretragu sadrži samo niz znakova bez blanko znaka; npr. za reč za pretragu `ispitati` u stringu `treba pitati pitanje za ispit` rezultujući string će biti `treba pitanje za`;
7. u stringu koji sadrži kôd na programskom jeziku C menja sva pojavljivanja poziva funkcije `calloc` pozivima funkciji `malloc`; uz uklanjanje suvišnih blanko znakova; npr. `a = (double *) calloc(n, sizeof(double));` se menja sa `a=(double*) malloc(n*sizeof(double));`

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 o grešci i prekinuti izvršavanje. Nakon svakog ciklusa programa (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.** Pre odbrane, priložiti na listu papira makar tri suštinski različita test primera i očekivane rezultate.

**Napomene:**

1. Odbrana trećeg domaćeg zadatka je u sredu, 15.05.2019., četvrtak, 16.05.2019. i petak, 17.05.2019. po rasporedu dostupnom na sajtu predmeta.
2. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):  **$i = (R + G) \bmod 8$**
3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
  - `dz3.c`, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;

07.05.2019. godine

*sa predmeta*