

PRAKTIKUM IZ PROGRAMIRANJA 2

- domaći zadatak broj 3 -

Sastaviti program na programskom jeziku C kojim se vrši određena vrsta obrade nad znakovnim nizom (u daljem tekstu, string). Program treba da repetitivno učitava sve potrebne ulazne podatke, ispisuje ih po učitavanju, izvršava zahtevanu obradu, ispisuje sve dobijene rezultate, i ponavlja navedene korake sve dok korisnik za ulazni string 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 počinje velikim slovom; npr. `empty_row_count` menja sa `emptyRowCount`;
1. radi inverznu izmenu od one iz tačke 0; svako veliko slovo u indentifikatoru osim početnog znači početak nove reči; npr. `RowCount` se menja u `Row_count`, `iADDFunc` u `i_a_d_d_func`;
2. slova u labeli menja u velika, izbacuje blanko simbole između labele i dvotačke i poravnava kod iza labele tako da počinje na dvanaestoj koloni; npr. `lab1 : add ax, bx` se menja sa `LAB1: add ax, bx`; labele nisu duže od 10 karaktera;
3. svaki string koji predstavlja for petlju napisanu na Pascal-u menja for petljom na jeziku C; npr. `for i:=100 downto a-b do` se menja sa `for(i=100; i>=a-b; i--)`; pretpostaviti da svaki string koji počinje rečju `for` predstavlja korektnu `for` naredbu;
4. ključne reči jezika Pascal (`For`, `To`, `Downto`, `Do`, `While`, `Repeat`, `Until`, `If`, `Then`, `Else`) menja tako da su sve ispisane velikim početnim slovom; takođe, ako je bilo koje od slova u navedenim ključnim rečima udvojeno, tu reč treba korigovati; npr. `whhille i<j DO` se menja u `While i<j Do`, `FORR i:=0 to 5 do` u `For i:=0 To 5 Do`;
5. koriguje brojeve u ulaznoj liniji tako da se znaci `!`, `@`, `#`, `$`, `%`, `^`, `&`, `*`, `(`, `)` unutar broja zamenjuju ciframa `1`, `2`, ..., `0`, respektivno; reč je broj ako počinje dekadnom cifrom. Ako reč sadrži slovo treba ga ukloniti; npr `1@` se menja sa `12`, `1(v4f` sa `194`;

Dimenzije stringa nisu poznate unapred. Memorija predviđena za string mora biti **dinamički alocirana**. Količina memorije koja je odvojena za string mora biti takva da iskorišćenje 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 string zadaje u jednom redu teksta putem standardnog ulaza (u string ulaze svi znakovi do `\n`, ne uključujući i `\n`). Obradu treba vršiti nad kopijom stringa, a pri ispisu rezultata na izlaz ispisati i neizmenjeni original i izmenjenu kopiju. Pre odbrane, priložiti na listu papira makar tri suštinski različita test primera i očekivane rezultate.

Važno: po uspešnom rešavanju osnovnog zadatka, modifikovati osnovni program tako zapamti svaki uneti red teksta i rezultat njegove obrade. Za smeštanje zapamćenih stringova koristiti dinamičkih niz stringova. Prilikom napuštanja programa, ispisati sve parove ulaznih stringova i rezultata njihove obrade.

Napomene:

1. Odbrana trećeg domaćeg zadatka je u nedelji od 08.05 do 10.05.2011. po rasporedu koji će biti blagovremeno dostupan 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): $i = (R + G) \bmod 6$
3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
 - `dz3.c`, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;
 - `dz3_mod.c`, koja sadrži izvorni tekst izmenjenog programa na programskom jeziku C;