

PRAKTIKUM IZ PROGRAMIRANJA 2

- domaći zadatak broj 2 -

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 se za ulazni string ne unese „dosta Brus Li“.

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

0. u stringu, koji sadrži i datume po formatu `dd-mm-gggg`, format datuma menja tako da umesto `mm` (koji označava mesec) stoji naziv meseca (04 se menja sa `April`), a ispred datuma se dodaje ime dana u sedmici kada pada taj datum;
1. u zadatom stringu, pronalazi reč/reči sa najvećim brojem istih samoglasnika, a zatim takva reč/reči izbacuje iz stringa;
2. vrši pretraga reči u stringu; reč za pretragu može sadržati džoker znak (?), koji menja tačno jedan znak; izbaciti iz stringa sve reči koje odgovaraju zadatoj;
3. utvrđuje da li string predstavlja tzv. *dancing* sekvencu, a zatim svi beli znakovi između dve reči zamenjuju jednim znakom razmaka; *dancing* sekvenca znakova je ona kod koje je prvo slovo veliko, a zatim se naizmenično smenjuju mala i velika slova (npr. `PrA kTi KuM`);
4. iz stringa uklanjaju sve one reči koje se pojavljuju više od jedanput, a na mesto prvog pojavljivanja takve reči upisuje ukupan broj pojava u stringu;
5. u stringu koji sadrži kod na nekom programskom jeziku sve pojave `"->"` u menjaju sa `"."`; konverziju ne treba vršiti ako se znakovi nalaze unutar komentara (`/*...*/`);
6. iz zadatog stringa izbacuje svako pojavljivanje nekog drugog stringa;
7. izbacuju iz stringa sve reči koje predstavljaju palindrom;

Predvideti maksimalnu dužinu stringa (`MAX_DUZ_STR`) pri deklaraciji stringa, a u toku rada proveravati da li je uneta dužina stringa takva da se svi elementi mogu smestiti u string predviđenih dimenzija. Smatra se da se string zadaje u jednom redu teksta koji se čita sa 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, student mora imati spremna makar tri suštinski različita test primera i očekivane rezultate.

Važno: kada se uspešno reši osnovni zadatak, modifikovati osnovni program tako da ne postoji ograničenje u dužini stringa koji može da se unese. Pošto dimenzije stringa nisu poznate unapred, memorija predviđena za string se mora dinamički alocirati. 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.

Napomene:

1. Termin odbrane drugog domaćeg zadatka studenti mogu videti na sistemu WebLab.
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:
 - **dz2.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;
 - **dz2_mod.c**, koja sadrži izvorni tekst izmenjenog programa na programskom jeziku C;
 - **dz2_test.txt**, koja sadrži test primere (ulaz i očekivani izlaz);