

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 datume u stringu zadate po američkom formatu `mm/dd/gg` ili `mm/dd/yyyy`, u evropski format `dd.mm.gggg` tako da umesto `mm` (što označava mesec) stoji naziv meseca (`04` se menja sa `April`), a ispred datuma se dodaje ime dana u sedmici kada pada taj datum; smatrati da je 01.01.1970. bio četvrtak, a da je godina prestupna ukoliko je ispunjen uslov `(god%4==0 && god%100!=0 || god%400==0)`;
1. menja sva vremena u stringu zadata po formatu `hh:mm:ss AM` ili `hh:mm:ss PM` iz 12-časovnog u 24-časovni oblik; `AM` označava prepodnevno vreme, a `PM` popodnevno vreme; npr. vreme `01:45:17 PM` se menja sa `13:45:17`;
2. u stringu koji sadrži kôd na nekom programskom jeziku menja sve pojave `."` koje se nalaze između dve reči sa `"->"`; konverziju ne treba vršiti ako se znakovi nalaze unutar komentara `(/*...*/)` ili ukoliko znaku `."` prethodi ili sledi blanko znak;
3. u stringu ispravlja sve greške u kucanju tipa `"Two INitial Caps"` (npr. `BAnana` promeniti u `Banana`), a zatim sve bele znakove između dve reči zamenjuje jednim blanko znakom;
4. u stringu ispravlja sve greške u kucanju tipa `"accidental cAPS LOCK usage"` (npr. `jABUKA` promeniti u `Jabuka`), a zatim sve bele znakove između dve reči zamenjuje jednim blanko znakom;
5. uklanja sve HTML tagove iz stringa; HTML tagovi su oblika `<tag>` ili `</tag>` (gde `tag` može biti bilo koji identifikator, npr. `body`, `head`, `title`), ali su važeći i oni koji sadrže proizvoljno belih znakova unutar zagrada kao npr. `< tag >` ili `< /tag>`;

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 14.05. do 16.05.2013. Tačni termini za sve studente su objavljeni 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;