

## 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 datume u stringu zadate po evropskom formatu **dd.mm.gggg** u američki format **Month dth, gggg**, tako da umesto **mm** (što označava mesec) stoji naziv meseca (**04** se menja sa **April**), a **dd** (što označava redni broj dana u mesecu) se zameni rednim brojem dana uz dodavanje odgovarajućeg sufiksa rednog broja (**st, nd, rd** ili **th**); npr. **Today is 08.05.2018.** se menja sa **Today is April 8th, 2018.**;
1. menja sva vremena u stringu zadata po formatu **hh:mm:ss** iz 24-časovnog u 12-časovni oblik uz dodavanje sufiksa **AM** koji označava prepodneвно vreme, ili **PM** koji označava popodneвно vreme; npr. **Alarm at 14:15:00** se menja sa **Alarm at 02:15:00 PM**;
2. uklanja iz stringa sav tekst koji ne predstavlja HTML tagove i ispisuje broj identifikovanih tagova; 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>**;
3. uklanja iz stringa sve one reči koje ne sadrže minimalan broj znakova iz zadate reči za pretragu; broj znakova zadaje korisnik; npr. za reč za pretragu **eto** u stringu **evo čuje se eho iz sobe** i minimalan broj znakova dva, rezultujući string će biti **evo eho sobe**;
4. vrši konverziju novčanih iznosa u stringu iz jedne valute u drugu; novčani iznos je realan broj nakon koga neposredno sledi oznaka valute; korisnik zadaje dva stringa koji predstavljaju oznake valute iz koje se konvertuje u valutu u koju se konvertuje i vrednost konverzije; npr. u stringu **Cena najma apartmana je 10.0 EUR tokom radne nedelje** se menja sa **Cena najma apartmana je 1180.0 RSD tokom radne nedelje**, ukoliko korisnik unese **EUR RSD 118.0**;
5. u stringu koji sadrži kôd na nekom programskom jeziku uklanja sve komentare; komentar se uvek nalazi između oznaka za početak i kraj komentara koje korisnik unosi sa standardnog ulaza (npr. oznake **/\*** i **\*/** Na programskom jeziku C ili oznake **{** i **}** programskom jeziku Pascal);
6. vrši sakrivanje svih telefonskih brojeva u zadatom stringu; telefonski broj obavezno počinje prefiksom **+381**, nakon čega slede brojevi koji mogu biti razdvojeni razmakom ili crticom; sakrivanje se vrši tako što se poslednje tri cifre zamene znakom **\***, a svi eventualni razmaci i crtice uklone; npr. string **Pozovi Petrovica na +381 64-555-333** se menja sa **Pozovi Petrovica na +38164555\*\*\***;
7. vrši ubacivanje emotikona („smajlija”) umesto odgovarajuće oznake koja se nalazi u zadatom stringu; u stringu se mogu naći oznake **:smesko** koja se menja sa **:)** **:ljutko** koja se menja sa **;-** (**:tugica** koja se menja sa **:-** (**:mangup** koja se menja sa **;-**) sekvencom; oznake emotikona obavezno počinju znakom dvotačka;

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 četvrtak, 17.05.2018. 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;

08.05.2018. godine

*sa predmeta*