

## PRAKTIKUM IZ PROGRAMIRANJA 2

### 2006-2007

- domaći zadatak broj 3 -

Sastaviti program na programskom jeziku C koji radi nad dinamički alociranim podacima na sledeći način:

- a) učitava sve potrebne podatke uz proveru validnosti (korektnosti) unetih podataka;
- b) vrši zahtevanu obradu nad ulaznim podacima;
- c) ispisuje sve dobijene rezultate;
- d) **ponavlja prethodne korake sve dok se ne zahteva kraj rada.**

U zavisnosti od grupe, treba uraditi sledeće:

0. Napisati potprogram koji utvrđuje da li je neki znakovni niz sastavni deo drugog znakovnog niza. Potprogram prima kao argumente pokazivače na dva znakovna niza, a kao rezultat vraća celobrojnu vrednost. Vrednost je negativna ako prvi niz nije pronađen u drugom nizu. U suprotnom, vrednost označava indeks u drugom nizu na kojoj poziciji je nađen prvi niz. U slučaju da se prvi niz nalazi više puta u drugom nizu, od interesa je isključivo prvo pojavljivanje. Napisati potprogram koji upoređuje dve reči, od kojih jedna može da sadrži "džoker" znake (\*), koji služe kao zamena za proizvoljan niz karaktera, proizvoljne dužine (što uključuje prazan niz, dužine 0). Potprogram prima pokazivače na dva znakovna niza, a kao rezultat vraća celobrojnu vrednost 0, ako uparivanje reči nije uspešno izvršeno, odnosno vrednost 1 ako jeste. Radi jednostavnijeg rešenja zadatka, preporučuje se da drugi potprogram koristi prvi. Glavni program treba da učitava niz rečenica sa standardnog ulaza (niz i rečenice su proizvoljne dužine), u kojima se ne pojavljuje "džoker" znak. Zatim učitava reč koja može da sadrži nijedan, jedan ili više "džoker" znaka. Konačno, sa sve unete reči bez "džoker" znakova, glavni program poziva potprogram za poređenje sa unetom reči koja može da sadrži "džoker" i na standardnom izlazu ispisuje sve one reči za koje je poređenje uspelo. Koristiti dinamičku alokaciju memorije.
1. Napisati potprogram koji utvrđuje da li je matrica A podmatrica matrice B. Kao rezultat, potprogram vraća pokazivač na celobrojni niz. Vrednost rezultata može biti NULL ako matrica A nije podmatrica matrice B. U suprotnom, prvi element vraćenog niza predstavlja broj parova vrsta-kolona koji označavaju poziciju gornjeg levog elementa na kojima se u matrici B nalazi matrica A. Napisati potprogram za učitavanje matrice sa standardnog ulaza. Potprogram pita korisnika za dimenzije matrice, alocira matricu i popunjava matricu vrednostima koje unosi korisnik. Kao rezultat, potprogram vraća pokazivač na blok memorije alociran za smeštanje matrice. Ako je rezultat NULL, matrica nije alocirana. Dimenzije matrice se vraćaju preko argumenata potprograma. Napisati potprogram za ispis matrice na standardnom izlazu. Napisati glavni program koji učitava matrice A i B, ispisuje njihov sadržaj a zatim pronalazi i ispisuje koordinate svih pojavljivanja matrice A u matrici B. Koristiti dinamičku alokaciju memorije.
2. Napisati program koji pomaže korisniku koji unosi neki tekst u otkrivanju pravopisnih grešaka. Program treba da preko standardnog ulaza pročita niz reči, proizvoljne dužine, koje su provereno tačne i koje predstavljaju rečnik. Zatim, program treba da omogući korisniku da unese neki tekst (niz karaktera proizvoljne dužine). Potom, program treba da upoređuje reč po reč iz unetog teksta sa rečima iz rečnika. U slučaju da ne pronađe sličnu reč u rečniku treba da ponudi korisniku mogućnost da pređe na sledeću reč ili da najpre datu reč doda u rečnik a zatim pređe na sledeću reč. U slučaju da pronađe sličnu reč (ili slične reči), osim mogućnosti pomenutih u prethodnom slučaju, treba još ponuditi mogućnost zamene date reči nekom od sličnih reči iz rečnika. U slučaju da pronađe identičnu reč u rečniku, automatski prelazi na sledeću reč ulaznog teksta. Na kraju program ispisuje korigovan tekst. Smatrati da su dve reči slične ako su im početna i krajnja slova ista, a njihove dužine se ne razlikuju za više od 1. Koristiti dinamičku alokaciju memorije. Unos reči za rečnik i rečenice za obradu realizovati u vidu potprograma. Pretragom teksta i dodavanje nove reči u rečnik realizovati kao posebne potprograme.

**Napomene:**

1. Studenti zadatke rešavaju **samostalno**. Rešenja zadataka **moraju** da sadrže navedene potprograme, a po potrebi ih može biti više nego što je navedeno. Potprogrami **ne smeju** da koriste globalne promenljive.
2. Za sve nedovoljno jasne zahteve u zadatku, studenti treba da usvoje **razumnu pretpostavku** u vezi realizacije programa. Prilikom odbrane, demonstratore treba obavestiti koja pretpostavka je usvojena (ili koje pretpostavke su usvojene).
3. Odbrana trećeg domaćeg zadatka će se obaviti tokom sedmice od 28. maja do 3. juna prema ranije objavljenom rasporedu.
4. Formula za računanje grupe problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):  
$$i = (R + G) \bmod 3$$
5. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati datoteke *dz3\_1.c* i *dz3\_2.c* koje sadrže izvorni tekst osnovnog programa na programskom jeziku C i koji može uspešno da se prevede. Uz datoteke dostaviti i nekoliko relevantnih test-primera.

21.05.2007.godine

*sa predmeta*