

Strukture podataka i algoritmi 1

Popravni kolokvijum

21. 06. 2019.

I kolokvijum

Na **Desktop-u** u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak11.c**, a rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak12.c**. Samo navedeni fajlovi biće pregledani. Samo zadaci koji se uspešno kompajliraju će biti bodovani, a broj poena će biti određen procentom ispravnih izvršavanja za svaki test primer.

1. Napisati funkciju **Izbaci** koja za prosleđeni string koji se sastoji samo od malih slova i crtica vraća string koji sadrži samo jedinstvena slova (bez ponavljanja) između svakog podstringa koji se nalazi između crtica.

Koristeći funkciju **Izbaci** napisati program koji najpre učitava broj linija teksta koje će biti učitane sa standardnog ulaza, a potom linije teksta (**jedna reč**) čija dužina neće prelaziti 100 karaktera, a potom svaki string ispisuje bez slova koja se ponavljaju.

Primer.

Ulaz	Izlaz
3 abba-afgg-ccc-fgh aaaa-bbbb-c abc-efgg-dd	ab-afg-c-fgh a-b-c abc-efg-d

Bodovanja. Zadatak vredi 12 poena ukoliko je ispis promenjenih linija nakon završenog učitavanja i ukoliko se koristi dinamička alokacija memorija. Korišćenje samo statički definisanih nizova umanjuje vrednost zadatka za 25%. Ispisivanje korigovanih linija nakon svake učitane linije umesto na kraju, takođe, umanjuje vrednost zadataka za 20%.

2. Napisati rekurzivnu funkciju **Okreni** koja za prosleđeni niz i indekse **i** i **j** treba da „okrene“ niz od indeksa **i** do indeksa **j**. Napisati funkciju **Ucitaj** koja učitava dimenziju niza i alokira i učitava niz celih brojeva. Napisati funkciju **Ispisi** koja za prosleđeni niz i dimenziju niza na standardni izlaz ispisuje niz. Napisati program koji koristeći navedene funkcije učitava dužinu niza A, potom niz celih brojeva, koji nakon toga „okreće“ čitav niz i ispisuje na standardni izlaz. Program potom učitava koliko elemenata treba dodati na niz A, učitava te elemente i „okreće“ samo dodate elemente i ispisuje ceo niz na standardni izlaz.

Primer.

Ulaz	Izlaz
6 1 2 3 4 5 6 3 7 8 9	6 5 4 3 2 1 6 5 4 3 2 1 9 8 7

Bodovanja. Zadatak vredi 10 poena ukoliko su korektno definisane sve date funkcije i pri tome nisu korišćeni nizovi fiksne dužina, već je memorijski prostor alociran dinamički. Ukoliko je funkcija **Okreni** definisana iterativno umesto rekurzivno, vrednost zadatka se umanjuje za 30%. Ukoliko su svi nizovi definisani statički vrednost zadatka se umanjuje za 20%. Ukoliko se učitavanje niza A realizuje u glavnom programu, a ne u funkciji, vrednost zadatka se umanjuje za 15%. Ukoliko se ne realizuje dodavanje novih elemenata i njihovo okretanja, vrednost zadatka se umanjuje za 20%.

II kolokvijum

Na **Desktop-u** u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak21.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak22.c**. Samo navedeni fajlovi biće pregledani. Samo zadaci koji se uspešno kompajliraju će biti bodovani, a broj poena će biti određen procentom ispravnih izvršavanja za svaki test primer.

1. Napisati program koji sadrži implementaciju rečnika uz pomoć povezanih lista. Ključ rečnika je **string**, a broj pojavljivanja je **ceo broj**. Napisati funkciju **Get** koja za prosleđeni ključ vraća pokazivač na element ukoliko postoji, odnosno NULL ako ne postoji. Napisati funkciju **Set** koja prihvata ključ i uvećava broj pojavljivanja za 1 elementu sa tim ključem, ukoliko takav element postoji, odnosno dodaje novi element na kraj rečnika ukoliko ne postoji. Napisati funkciju **Form** koja treba da učitava **ceo broj n** i **n reči** i formira i ispisuje rečnik. Nakon toga, izbaciti sve reči koje se pojavljuju samo jednom i ispisati rečnik.

Ulaz:

10
kuglica
posta
kuglica
tastatura
auto
leptir
kuglica
tastatura
vetar
list

Izlaz:

kuglica 3
posta 1
tastatura 2
auto 1
leptir 1
vetar 1
list 1

Nakon izbacivanja:

kuglica 3
tastatura 2

Napomena. Zadatak nosi **14** poena. Liste je dozvoljeno kreirati bilo dodavanjem elemenata na početak, bilo na kraj liste. Ukoliko elementi liste ne sadrže informaciju o broju pojavljivanja reči, već samo reči, vrednost zadatka se umanjuje za 20%. Ukoliko se rečnik kreira u glavnom delu programa, vrednost zadatka se umanjuje za 15%. Ukoliko se ne izbacе reči koje se ne ponavljaju, vrednost zadatka se umanjuje za 20%.

2. Na društvenu mreži IMIVibe korisnici se prijavljuju preko jedinstvenog imena. Mreža je organizovana po interesnim grupama. Svaki korisnik može da se prijavi na više različitih grupa i svaka grupa ima više različitih korisnika.

a) Definirati strukturu **parovi** koja pamti par ime korisnika-grupa na koju je prijavljen i strukturu **korisnik** koja za svakog korisnika pamti njegovo ime, na koliko grupa je prijavljen i niz grupa na koje je prijavljen.

b) Napisati funkciju **UnosKorisnika** koja iz datoteke sa nazivom **IMIVibe.txt** parove korisničko ime i grupa na koju je korisnik prijavljen i formira niz struktura **parovi**. U prvoj liniji datoteke nalazi se broj parova u datoteci. Isti par korisnik-grupa neće se pojaviti više puta.

c) Napisati funkciju **Transform** koja od niza parova formira niz korisnika, gde se svaki korisnik javlja jednom sa podacima o grupama kojim pripada.

d) Napisati funkciju **main** koja učitava parove korisnika i grupa i ispisuje dobijeni niz. Zatim formira niz korisnika i ispisuje ga.

IMIVibe.txt	Izlaz

Napomena. Zadatak nosi **14** poena. Ukoliko se podaci ne učitavaju iz datoteke, već sa tastature, vrednost zadatka se umanjuje za 10%. Realizacija fukcije **Transform** predstavlja 50% vrednosti zadatka. Ukoliko ne niz grupa kojima korisnik pripada ne realizuje preko niza pokazivača, vrednost zadatka se umanjuje za 15%.