

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Algoritmi i strukture podataka 1 (13S111ASP1)  
*Nastavnici:* dr Milo Tomašević, red. prof.; doc. dr Marko Mišić  
*Asistent:* Sanja Delčev, dipl. ing.; Maja Vukasović, dipl.ing.;  
Dragana Milovančević, dipl.ing.  
*Ispitni rok:* Prvi kolokvijum (mart 2019.)  
*Datum:* 19.03.2019.

*Kandidat\*:* \_\_\_\_\_

*Broj Indeksa\*:* \_\_\_\_\_

*Kolokvijum traje 120 minuta. Prvih 60 minuta od početka nije dozvoljeno napuštanje sale.  
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /5	<i>Zadatak 5</i>	_____ /15
<i>Zadatak 2</i>	_____ /15	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 3</i>	_____ /15	<i>Zadatak 7</i>	_____ /15
<i>Zadatak 4</i>	_____ /15	<i>Zadatak 8</i>	_____ /10

**Ukupno na kolokvijumu:** \_\_\_\_\_ /100

**Napomena:** Ukoliko u postavci nekog zadatka postoje nepreciznosti, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

\* popunjava student.

---

1. [5] Neka se u nekom programu za generisanje 3-bitnih pseudoslučajnih neoznačenih celih brojeva koristi BBS generator (*Blum, Blum, Shub*). Parametar generatora  $M$  ima vrednost  $19 \cdot 11 = 209$ . Za klicu je uzeta vrednost  $X_0 = 10$ , a za formiranje sekvence brojeva uzima se najniži bit rezultata. Odrediti prva tri pseudoslučajna broja koje će generator generisati.

2. [15] Posmatra se retka matrica predstavljena ulančanim kružnim listama sa zaglavljem. Dati su nizovi  $R$  i  $C$  sa svojim dužinama  $n$  i  $m$ . Broj nenultih elemenata je  $k$ . Predstaviti datu matricu pomoću jednog vektora zapisa sa po tri polja.

TRANSFORM( $R, C, n, m, k$ )

3. [15] Posmatra se vektorska implementacija jednostruko ulančane liste celih brojeva u vektoru  $L[1:n]$ . Napisati u pseudokodu funkciju koja iz liste  $L$  efikasno uklanja element čiji je indeks smešten u promenljivu *old*, pod pretpostavkom da element koji se uklanja nije poslednji element liste. Operacija treba da ažurira i listu slobodnih ulaza. Promenljiva *avail* pamti indeks prvog elementa u listi slobodnih ulaza.

LIST REMOVE( $L, old, avail$ )

4. [15] Dat je red u kome su smešteni celi brojevi. Realizovati operacije dodavanja celobrojne vrednosti u strukturu (*add*) i dohvatanja celobrojne vrednosti iz strukture (*get*) tako da se pri dohvatanju vrednosti iz ovoga reda favorizuju neparni brojevi u odnosu na parne. Dozvoljena je upotreba još jedne pomoćne linearne strukture podataka (steka ili reda). Njima je moguće pristupati samo upotrebom funkcija *push/pop* i *insert/delete* koje su dostupne kao gotove funkcije. Međusobni poredak u okviru klasa brojeva (parni i neparni) ne treba menjati.

ADD( $queue, num$ )

GET( $queue$ )

5. [15] Napisati u pseudokodu funkciju koja korišćenjem steka proverava da li niz *arr1* predstavlja stek permutaciju niza *arr2*. Jedan niz predstavlja stek permutaciju drugog niza ukoliko se može dobiti transformacijom koja uključuje korišćenje samo operacija sa stekom i jedan prolazak kroz polazni niz. Elementi se na stek isključivo mogu stavljati iz niza *arr1*, a sa steka se isključivo mogu stavljati u niz *arr2*. Nizovi su iste dužine.

STACK PERMUTATION(*arr1*, *arr2*, *len*)

6. [10] Neka se na nekom namenskom računarskom sistemu vrši očitavanje N odbiraka sa senzora pritiska i rezultati smeštaju u niz  $A[1:N]$ . Jedan odbirak može uzeti vrednost u opsegu od 0 do 99. Objasniti na koji način se odbirci efikasno mogu smestiti u niz  $A$  i izvesti adresnu funkciju za pristup proizvoljnom elementu niza  $A$ . Širina jedne memorijske reči je 32 bita.

Adresna funkcija:



8. [10] Neka se posmatra implementacija prioritelnog reda u vidu uređene, jednostruko ulančane liste. Red sadrži uređene parove (prioritet, simbol), a manja vrednost označava veći prioritet. Objasniti na koji način se kod ove implementacije obezbeđuje FIFO poredak dohvaćanja elemenata istog prioriteta. Na primeru umetanja elemenata (14, A), (3, C), (7, T), (7, X), (3, B), (1, Y) i (5, D), dohvaćanja četiri elementa i umetanja elemenata (4, G) i (7, E) prikazati stanje reda po koracima.