



## CS203 - DOMAĆI ZADATAK 06.

Prilikom slanja domaćeg zadatka svom asistentu neophodno je da ispunite sledeće:

- Subject mail-a mora biti CS203-DZbr. Za ovaj domaći zadatak - CS203-DZ06
- Sve fajlove, koji su deo rešenja zadatka, arhivirati (zip)
- **Obavezno kopirati tekst zadatka kao komentar u kodu resenja**
- U prilogu maila treba da se nalazi arhiva projekta koji se ocenjuje, imenovana na sledeći način: CS203-DZbr-BrojIndeksa-ImePrezime.

*Na primer, CS203-DZ06-9999-VeljkoGrkovic.zip*

**Molimo sve studente da se pridržavaju navedenog, inače zadaci neće biti pregledani i ocenjeni.**

Studenti iz Beograda zadatke na ocenjivanje šalju mail-om na adresu

[lazar.mrkela@metropolitan.ac.rs](mailto:lazar.mrkela@metropolitan.ac.rs)

Studenti iz Nisa zadatke na ocenjivanje šalju mail-om na adresu

- IT smer: [jovana.jovanovic@metropolitan.ac.rs](mailto:jovana.jovanovic@metropolitan.ac.rs)
- SI smer: [uros.lazarevic@metropolitan.ac.rs](mailto:uros.lazarevic@metropolitan.ac.rs)

Online studenti zadatke na ocenjivanje šalju mail-om na adresu

[uros.lazarevic@metropolitan.ac.rs](mailto:uros.lazarevic@metropolitan.ac.rs)

**Rok za slanje domaćih zadataka:**

- **Tradicionalni studenti: sedam dana od dana održavanja vežbe na koju se odnosi domaći zadatak. Nakon toga, broj poena se umanjuje za 50%. Krajnji rok za slanje rešenja je deset dana pre termina ispita u kome želite da polažete ispit.**
- **Studenti na online nastavi, kao i studenti kojima je predmet prenet u narednu godinu ili je diferencijalni: najkasnije deset dana pre termina ispita u kome želite da polažete ispit.**

Svaki student radi jedan zadatak sa spiska. Ostali zadaci mogu da posluže za vežbanje i pripremu ispita, ali ih ne šalžete na pregled asistentima. Redni broj zadatak dobijate tako što vaš broj indeksa podelite sa 12, a dobijeni ostatak pri deljenju uvećate za 1. Na primer: **Broj indeksa 2378**

**2378 % 12 + 1 = 3 – student radi 3. zadatak.**

1. Napišite program koji koristi bucket sort za sortiranje studenata prema njihovim ocenama. Svaki student će imati ime, prezime, indeks i ocenu, a program bi trebalo da koristi sortiranje po njihovim ocenama od najniže do najviše i da rezultat odštampa kao listu studenata.
2. Napišite program koji koristi bucket sortiranje za sortiranje niza decimalnih brojeva između 0 i 1. Korisnik će uneti niz decimalnih brojeva i program će ih sortirati pomoću od najmanjeg do najvećeg. Nakon sortiranja decimalnih brojeva, program treba da izračuna aritmetičku sredinu (prosek) svih sortiranih brojeva i da je odštampa kao dodatni izlaz.
3. Neka imamo listu studenata, pri čemu svaki student ima ime, prezime, indeks i datum rođenja. Napišite program koji koristi radix sort za sortiranje ovih studenata prema godinama rođenja od najstarijeg do najmlađeg.
4. Napisati program koji koristi radix sort za sortiranje niza pozitivnih celih brojeva, ali tako da program prati i broj poređenja koja se vrše prilikom sortiranja. Korisnik treba da unese niz brojeva, a program treba da kao rezultat odštampa sortirani niz brojeva od najmanjeg do najvećeg, kao i broj poređenja koja su izvršena .
5. Napisati program koji koristi heap sort za sortiranje niza pozitivnih celih brojeva, a zatim omogućava korisniku da unese broj koji želi da pronađe u sortiranom nizu. Program treba da odštampa indeks pronađenog broja (ako je pronađen) ili poruku da broj nije pronađen.
6. Napisati program koji koristi heap stablo za čuvanje celih brojeva. Program treba da omogući korisniku da unese brojeve, a zatim omogući sledeće operacije:
  - Dodavanje novog broja u stablo
  - Štampanje minimalnog elementa
7. Napišite program koji koristi bilo koji algoritam za sortiranje (quick sort, merge sort ili bilo koji drugi) za sortiranje niza stringova i uklanjanje duplikata. Korisnik će uneti niz nizova, a program će ih sortirati po abecednom redu od A do Z i ukloniti sve duplikate.
8. Napišite program koji koristi algoritam sortiranja mehurića (bubble sort) za sortiranje niza stringova. Nakon sortiranja, program treba da proveriti svaki string i odštampa samo one koji počinju slovom "A" ili "B".
9. Napišite program koji prima tekst od korisnika, a zatim sortira reči u tekstu tako da proverava da li su reči anagrami. Nakon sortiranja, program treba da odštampa sve grupisane anagrame ( u listama), dok ostale reči zanemaruje.  
*Ulaz: monitor jabuka mis muzika oimnort ims*  
*Izlaz:*  
*monitor, oimnort*  
*ims, mis*

10. Napišite program koji koristi bucket sort za sortiranje niza pozitivnih celih brojeva. Pre sortiranja, program treba da proveriti da li su svi brojevi u nizu prosti. Ako nisu, program bi trebalo da odštampa poruku o tome. Ako su svi brojevi prosti, program bi trebalo da ih sortira i odštampa sortirani niz.
11. Napisati program koji koristi heap stablo za čuvanje celih brojeva. Program treba da omogući korisniku da unese brojeve, a zatim omogući sledeće operacije:
- Dodavanje novog broja u stablo
  - Brisanje maksimalnog elementa
12. Napišite program koji koristi radix sort za sortiranje niza pozitivnih celih brojeva. Pre sortiranja, program treba da proveriti da li su svi brojevi u nizu palindromi (tj. čitaju iste s leva na desno i s desna na levo). Ako nisu, program bi trebalo da odštampa poruku o tome. Ako su svi brojevi palindromi, program treba da ih sortira i odštampa sortirani niz.