

Ime i prezime	Broj indeksa

Obavezan deo A (15 poena – obavezno 10 poena):

- (2 poena) Iz date liste celih brojeva korišćenjem **filtriranja listi** formirati novu listu koja sadrži samo parne brojeve prve liste.

brojevi = [5,8,1,3,8,9,22,28,34,74,11,2,1]
parni_brojevi = _____
- (3 poena) Napisati funkciju **Cifre** u Python-u koja prihvata listu celih brojeva i vraća rečnik čiji elementi imaju formu `cifra:broj_pojavljivanja`. (Za sve cifre koje su se javile u zapisu brojeva liste odrediti koliko puta su se ukupno javile – broj_pojavljivanja)
- (3 poena) Napisati rekurzivnu funkciju `dupliranS(S)` kojom se proverava da li se string `S` sastoji od dva ista nadovezana dela. (na pr. `dupliranS("lepolepo")` -> `true`, `dupliranS("dannoc")` -> `false`)
- (3 poena) Napisati funkciju koja određuje sve proste brojeve manje od zadanog broja `n` koristeći algoritma „Eratostenovo sito“.
- (4 poena) Odrediti poslednji element `n`-tog reda trougaonog ispisa (na pr. `n=3` -> 57338)? (Izvedena opšta formula tog elementa donosi 4 poena, u suprotnom zadatak nosi 2 poena)

1 8
22 50 106 218
442 890 1786 3578 7162 14330 28666 57338
.
.

Izborni deo B (15 poena):

1. Svako od zaposlenih u IMI TEX kompaniji dnevno dobija šifrovan tekst koji obrađuje. Obrađivanje se svodi na prepoznavanje svih reči koje su se javile u tekstu i određivanja broja pojavljivanja svake od njih. Kako su reči Spa i SPA iste, sve pročitane reči pamtiti kao da su napisane malim slovima. Podrazumeva se da reči mogu da sadrže samo slova (više od jednog slova, a graničnici između reči mogu biti bilo koji karakteri koji nisu slova). na primer: ako tekst sadrži Ab#c, u tom slučaju reč je samo ab.

U fajlu **Obrada.xml** dati su podaci o dnevnim šifrovanim tekstovima koje je dobio svaki od radnika (pogledati primer fajla).

Napisati program koji vrši organizaciju obrađenih tekstova na sledeći način:

- Za svakog radnika se pamte njegove reči i koliko puta se svaka od njih javila.
- U fajl **SredjenTekst.json** za svakog radnika ispisati njegove reči i njihov broj pojavljivanja (pogledati primer fajla).
- Za svakog radnika odrediti koja se reč u okviru njegovog šifrovanog teksta javila najviše puta.
- Odrediti listu reči koje su se javile kod svih radnika.

Obrada.xml	SredjenTekst.json
<pre><xml> <obrada> <radnik>Marko</radnik> <tekst> Sifra za racunar u racunarskoj Sali je student#pmf I mogu da se loguju svi student PMF-a. Racunar Pmf da se *a*AB. Svi &ab/da! </tekst > </obrada> <obrada> <radnik>Nenad</radnik> <tekst> DaNAs \$jE mrak!Danas student*je*NA pmf-u. </tekst > </obrada> <obrada> <radnik>Ana</radnik> <tekst> Sifra za racunar u racunar%% je student.Svi &ab/da! </tekst > </obrada> </xml></pre>	<pre>[{ Marko:{sifra:1, za:1, racunar:2, racunarskoj:1, sali:1, je:1 student:2 pmf:3 mogu:1 da:3 se:2 loguju:1 svi:2 ab:2}, } { Nenad:{danas:2, je:2, mrak:2, student:1, na:1, pmf:1 }, } { Ana:{sifra:1, za:1, racunar:2, je:1, student:1, svi:1 ab:1 da:1}, }]</pre>