

1. **[Haskell 34%]** Napisati funkciju koja od zadate liste od k podlisti brojeva formira listu brojeva u kojoj k-ti broj predstavlja zbir svih prostih elemenata k-te podliste. Implementacija treba da koristi sve tri popularne funkcije višeg reda: filter, map i fold(l,r). Za implementaciju provere da li je broj prost koristiti lambda izraz u kombinaciji sa set comprehension tehnikom. Implementacija koja ne koristi funkcije višeg reda, lambda izraze i set comprehension tehniku može da osvoji najviše 28%. Test primer:

**h1** [[3,4,5,3,2],[2,2,3,5,76,1,3],[13,19,2,6,1]] → [13,15,34]

2. **[Prolog 33%]** Napisati prolog predikat koji zadatu listu brojeva razbija na dve liste takve da razlika suma brojeva u te dve liste bude što manja – ne mora optimalno da se reši, samo ne sme da bude ni „previše loše“ (**Jedna ideja: znamo da razbijemo rep rekurzivno na takve dve liste, ostaje pitanje šta ćemo sa glavom liste?**). Dozvoljeno je koristiti pomoćne parametre u okviru predikata. Test primer (Vaš rezultat ne mora biti kao u test primeru):

**p2**([5,3,6,3,3,1,3,4,3,1,4,6,3],X,Y):-  
**X**=[3,4,3,3,3,3,5], **Y**=[6,1,4,1,3,6]

3. **[Java Spark 33%]** U datoteci reci.txt koja se nalazi u direktorijumu “11/reci.txt” se nalaze četvoroslovne niske od po 4 karaktera. Potrebno je grupisati sve reči prema sumi ascii karaktera i prebrojati pojavljivanja, i nakon toga ispisati grupisanje u formi: suma broj\_pojavlivanja uređeno opadajuće prema broju pojavljivanja. Npr. suma(EBXI) = 69+66+88+73=296, suma(QHJH)=81+72+74+72=299, suma(JHHQ)=299, itd. Test primer:

EBXI  
JHHQ  
EBXI  
IXBE  
BBEE  
HHQJ

- 296 3
- 299 2
- 270 1