Programske paradigme – grupa A

1. **[Haskell 34%]** Ulazni argument funkcije je matrica predstavljenja kao lista listi (pretpostaviti da je matrica kvadratna). Napisati funkciju koja računa sumu neparnih elemenata koji se nalaze striktno iznad glavne dijagonale.Test primer:
   h1 [[1,2,**3**,4],[5,6,**7**,8],[9,10,11,12],[0,1,2,5]] = 10

2. **[Prolog 33%]** Predikat prihvata dve liste dužine N, od kojih jedna predstavlja niz vrednosti, a druga permutaciju (niz indeksa) u kojoj se ta lista vrednosti trenutno nalazi. Dovesti listu do originalnog uređenja, odnosno uređenja koje odgovara permutaciji [1,2,…,N] (**ideja za rešavanje**: sortirati istovremeno i niz indeksa i niz vrednosti, tj. svaki put kada se indeks iz niza indeksa pomera, pomera se i vrednost na istoj poziciji u nizu vrednosti).
   p1([-353,33,356,3,14,75,3,-3], [2,5,1,3,8,4,7,6],X):-
   X=[356,-353,3,75,33,-3,3,14]

3. **[Java Spark 33%]** Napisati distribuirani program koji vraća sumu ASCII vrednosti svih reči kod kojih ASCII vrednosti slova formiraju strogo rastuću ili strogo opadajuću sekvencu. Npr. reč ABCD i VGDA ispunjavaju taj uslov, dok reči AABD i DBAG ne ispunjavaju. Pod validnim rečima uzimati u obzir samo one koje su zapisane u potpunosti velikim slovima. Testirati nad datotekom reci.txt koja se nalazi u vezbe/11 direktorijumu, kao i nad ovim manjim test primerom:
   ABCD VGDA AABD DBAG dfeb abcd **ABEF**
   826

Napomena: Haskell funkciju i Prolog predikat imenovati sa h1 odnosno p1 kao što je gore navedeno i sačuvati ih u datotekema pod nazivom h1.hs odnosno p1.pl. Za Java Spark zadatak se ostavlja samo paket u kojem se nalazi kod. Ove dve datoteke i paket direktorijum za Spark ubaciti u direktorijum koji se zove GrupaAmiXXXYYImePrezime gde je miXXXYY zamenjeno sa oznakom naloga na alas-u. **Vreme za rad:** (**2 sata 45 minuta – vreme potrošeno na teorijskom delu).**

Programske paradigme – grupa A

1. **[Haskell 34%]** Ulazni argument funkcije je matrica predstavljenja kao lista listi (pretpostaviti da je matrica kvadratna). Napisati funkciju koja računa sumu neparnih elemenata koji se nalaze striktno iznad glavne dijagonale.Test primer:
   h1 [[1,2,**3**,4],[5,6,**7**,8],[9,10,11,12],[0,1,2,5]] = 10

2. **[Prolog 33%]** Predikat prihvata dve liste dužine N, od kojih jedna predstavlja niz vrednosti, a druga permutaciju (niz indeksa) u kojoj se ta lista vrednosti trenutno nalazi. Dovesti listu do originalnog uređenja, odnosno uređenja koje odgovara permutaciji [1,2,…,N] (**ideja za rešavanje**: sortirati istovremeno i niz indeksa i niz vrednosti, tj. svaki put kada se indeks iz niza indeksa pomera, pomera se i vrednost na istoj poziciji u nizu vrednosti).
   p1([-353,33,356,3,14,75,3,-3], [2,5,1,3,8,4,7,6],X):-
   X=[356,-353,3,75,33,-3,3,14]

3. **[Java Spark 33%]** Napisati distribuirani program koji vraća sumu ASCII vrednosti svih reči kod kojih ASCII vrednosti slova formiraju strogo rastuću ili strogo opadajuću sekvencu. Npr. reč ABCD i VGDA ispunjavaju taj uslov, dok reči AABD i DBAG ne ispunjavaju. Pod validnim rečima uzimati u obzir samo one koje su zapisane u potpunosti velikim slovima. Testirati nad datotekom reci.txt koja se nalazi u vezbe/11 direktorijumu, kao i nad ovim manjim test primerom:
   ABCD VGDA AABD DBAG dfeb abcd **ABEF**
   826

Napomena: Haskell funkciju i Prolog predikat imenovati sa h1 odnosno p1 kao što je gore navedeno i sačuvati ih u datotekema pod nazivom h1.hs odnosno p1.pl. Za Java Spark zadatak se ostavlja samo paket u kojem se nalazi kod. Ove dve datoteke i paket direktorijum za Spark ubaciti u direktorijum koji se zove GrupaAmiXXXYYImePrezime gde je miXXXYY zamenjeno sa oznakom naloga na alas-u. **Vreme za rad:** (**2 sata 45 minuta – vreme potrošeno na teorijskom delu).**