UVOD U ALGORITME

Uvod u programski jezik Julia – čas 01

Cilj lekcije

- Cilj lekcije je da studenti savladaju osnovne elemente programskog okruženja i jezika Julia.

- Očekuje se da student nakon savladavanja lekcije može uspešno da rukuje matricama i vektorima, u smislu njihovog generisanja, spajanja, pristupa elementima, korišćenja osnovnih ugrađenih funkcija, itd.

Neophodna predznanja

Glavna i sporedna dijagonala matrice

- Glavnu dijagonalu neke matrice A veličine nxn sa elementima $a_{i,j}$, $i \in \{1..n\}$, $j \in \{1..n\}$ čine elementi $a_{i,i}$, $i \in \{1..n\}$. Sporednu dijagonalu čine elementi $a_{i,n-i+1}$, $i \in \{1..n\}$.

Jedinična matrica

- I jedinična matrica je matrica koja na glavnoj dijagonali ima sve jedinice, a ostali elementi su jednaki nuli.
- Primer jedinične matrice 3x3 :

1	0	0
0	1	0
0	0	1

Neophodna predznanja

Inverzna matrica

- Inverzna matrica kvadratne matrice A je matrica A^{-1} za koju važi da je:

  $AA^{-1} = A^{-1} A = I$

- Pronalaženje inverzne matrice je važan problem u matematici i postoji više numeričkih algoritama za njegovo rešavanje.

Neophodna predznanja

Množenje matrica

- Množenje matrica je različita operacija od množenja odgovarajućih elemenata matrice.
- Primer množenja matrica dimenzija 2x2:

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 1*2+2*5 & 1*2+2*1 \\ 0*2+3*5 & 0*2+3*1 \end{bmatrix} = \begin{bmatrix} 12 & 4 \\ 15 & 3 \end{bmatrix} \neq \begin{bmatrix} 1*2 & 2*2 \\ 0*5 & 3*1 \end{bmatrix}$$

- Kod množenja matrica ne važi osobina komutativnosti:

$$AB \neq BA$$

Deljenje matrica

- Deljenje matrica se vrši na sledeći način: $A / B = A * B^{-1}$

Neophodna predznanja

Sistem linearnih algebarskih jednačina i matrice

- Rešavanje sistem linearnih algebarskih jednačina (SLAJ) se svodi na problem rešavanja matrične jednačine:

  $Ax = b$

- Iz toga sledi da je vektor rešenja:

  $x = A^{-1}b$

Neophodna predznanja

Camel notation

- Kamel notacija (Camel notation) je praksa u pisanju složenih reči gde svaka sledeća reč počinje sa velikim slovom, dok je početno slovo malo, na primer: prviPodatak, novaGodina, prviMaj, noviSad, noviSadSpens, itd.

Case sensitivity

- Za programski jezik kažemo da je case-sensitive ukoliko razlikuje mala i velika slova kod imenovanja promenljivih (npr. Niz i niz su 2 različite promenljive).

Julia

- Julija je programski jezik opšte namene
- Veoma je popularan u oblastima:
- numeričke analize, scientific computing, u inženjerstvu, mašinskom učenju, računarskim naukama, statistici.
- Veoma je sličan jezicima: MATLAB, R, Python, Octava, a po brzini izvršavanja poredi se sa C/C++

Osobine:

- Veoma dobre performanse
- Sadrži dinamičke tipove i omogućava jednostavan interaktivni rad
- Podržava multiple dispatch
- Jezik je visokog nivoa i lako se uči: radi sa vektorima, matricama, ...
- Sadrži proširenu standardnu biblioteku i brojne pakete
- Besplatan i open source softver

Julia - Instalacija

- Moguće je preuzeti putem sledećeg linka: https://julialang.org/downloads/
- U slučaju da koristite Linux OS, za verziju 1.2.0
  - *odaberite: Generic Linux Binaries for x86 64-bit (ili 32-bit)*
  - *Nakon preuzimanja raspakujte arhivu, u terminalu to možete učiniti komandom: tar -xvzf naziv_arhive.tar.gz*
  - *Premestite raspakovan direktorijum u /opt/ direktorijum: sudo mv naziv_foldera /opt/*
  - *Nakon ovoga potrebno je dodati Juliu na putanju kako bi bila dostupna iz svakog foldera, jedan od načina je da se u okviru .bashrc datoteke(ukoliko ne postoji napraviti je komandom touch .bashrc u okviru home direktorijuma) na kraj dodaju sledeće linije: export JULIA=/opt/julia-1.2.0 i export PATH=$PATH:$JULIA/bin , nakon ovoga u terminalu pozvati source ~/.bashrc kako bi se promene primenile i na postojeću sesiju*
- **Analogan postupak je i za novije verzije**

Julia - Pokretanje

- Nakon instalacije Juliu je moguće pokrenuti tako što se klinke na ikonu Julia
  - *pod Linux-om u terminalu ukucati julia*
- Podržan je interaktivni rad gde se koristi REPL (Read–Eval–Print Loop). Izraz koji se ukuca se odmah izvršava i prikazuje se rezultat.
- Julia pamti sve prethodno izračunate vrednosti u svom okruženju!!!
- Direktorijum u kom smo bili prilikom pozivanja *julia* komande iz terminala predstavlja trenutni radni direktorijum
- Radni direktorijum je moguće promeniti naredbom cd(path), na primer:
  - *cd("c://UvodUAlgoritme")*
- Prelazak u režim pomoći (help) vrši se tasterom ? a zatim se otkuca traženi pojam

Julia - Operator dodele

- Za dodelu vrednosti nekoj promenljivoj koristi se operator ′=′. Npr. ako želimo da dodelimo vrednost 1 promenljivoj z, u komandni prompt pišemo naredbu: z = 1. Ako je promenljiva z imala vrednost pre dodele, ta vrednost se briše.

- Primetiti da je "=" dodela vrednosti gde se vrednost izračunate desne strane (tj. iskaza) dodeljuje promenljivoj navedenoj sa leve strane znaka "=". Pri ovome nije navedeno kog tipa treba da bude promenljiva, nego je na osnovu tipa rezultata izračunavanja iskaza određen i tip promenljive.

- Nije potrebno deklarisati promenljive pre nego što im se prvi put dodeli vrednost!!!

Julia - Aritmetički operatori

Izraz	Ime	Opis
`+x`	Unarni plus	Operacija identiteta
`-x`	Unarni minus	Preslikave vrednosti u njihove aditivne inverzne vrednosti
`x + y`	binary plus	Sabiranje. Ako su oba operanda matrice, dimenzije matrica moraju biti iste.
`x - y`	binary minus	Oduzimanje. Ako su oba operanda matrice, dimenzije matrica moraju biti iste.

Julia - Aritmetički operatori

Izraz	Ime	Opis
x * y	množenje	Množenje matrica. Broj kolona od matrice x mora biti jednak broju vrsta od matrice y.
x / y	deljenje	Desno deljenje, ekvivalentno izrazu x*inv(y)
x ÷ y	Celobrojno deljenje	x / y, zaokruženo na cele brojeve
x \ y	Obrnuto levo deljenje	Levo deljenje, ekvivalentno izrazu inv(x)*y
x ^ y	Stepenovanje	Računa x na stepen y

Julia - Operatori poređenja

Izraz	Opis
==	True ako je x jednako y
!=, ≠	True ako je x različito od y
<	True ako je x manje od y
<=, ≤	True ako je x manje ili jednako y
>	True ako je x veće od y
>=, ≥	True ako je x veće ili jednako y

Julia - Logički operatori

Izraz	Opis
`!x`	Logičko NE. Negacija. Rezultat je true ako je element false i obratno
`x && y`	Logičko I. Rezultat je true samo ako su oba elementa true
`x || y`	Logičko ILI. Rezultat je true ako je barem jedan od elemenata true

Julia - Operator tačke

- Za svaku binarnu operaciju postoji njoj korespodentna operacija sa operatorom tačke.
- Kada se ispred binarnog operatora navede oprator tačke, npr .+ to označava da se operacija vrši nad pojedinačnim elementima (elementwise). Tako, na primer, operacija stepenovanja vektora nema smisla [1, 2, 3]^3, ali operacija stepenovanja svakog od elemenata vektora ima: [1, 2, 3] .^ 3 daje za rezultat [1, 8, 27]

- Primer: Ako su date matrice A = [1 2; 3 4] i B = [1 1; 1 1], tada je:

  *A*B = [3 3; 7 7]*

  *A.*B = [1 2; 3 4]*

Julia - Primeri ugrađenih funkcija

sin(X), cos(X), tan(X)	Vrednost sinusa (kosinusa, tangensa) svakog elementa matrice (vektora) u radijanima.
sqrt(X)	Koren svakog elementa matrice (vektora). Ako element ima negativnu vrednost, vraća kompleksan broj.
abs(X)	Intenzitet kompleksnog broja
exp(X)	Izračunava x e svakog elementa iz matrice (vektora) X
log(X), log10(X)	Prirodni logaritam i logaritam sa osnovom 10 od svakog elementa matrice (vektora) X.
rem(X,Y)	Ostatak pri deljenju X/Y, izračunat preko izraza x-y.*fix(x./y)
round(X)	Vraća integer najbliži elementu X. Ako je X matrica, vraća matricu integera najbližih elementima u X.
floor(X) ceil(X)	Vraća najveći (najmanji) integer koji nije veći (manji) od X. Ako je X matrica, vraća matricu najvećih (najmanjih) integera koji nisu veći (manji) od elemenata u matrici X.
factorial(X)	Vraća faktorial prirodnog broja X. X! = 1*2*3*...*X

Julia - Primeri ugrađenih funkcija

- U slučaju matrica, da bi neka od prethodno navedenih funkcija bila primenjena na svaki od elemanata matrice neophodno iza naziva funkcije staviti operator tačke, npr: sin.([1 2; 3 4; 5 6]) što daje sledeći rezultat:

sin.([1 2; 3 4; 5 6]) =

0.841471	0.909297
0.14112	-0.756802
-0.958924	-0.279415

Julia - Predefinisane konstante

- Predefinisane konstante se nalaze u paketu: Base.MathConstants
- Neophodno ih je uključuti pre upotrebe naredbom: using Base.MathConstants
Ili navođenjem punog imena: Base.MathConstants.pi , Base.MathConstants.e, itd.
- **Primer 1.1:** Napisati u Juli-i sledeće izraze:

a) $\frac{log_{10}100}{log_{10}10}$

b) $\left\lfloor \frac{1+cos\frac{\pi}{4}}{1.2} \right\rfloor$

Julia - Primer

- **Primer 1.1:** Napisati u Juli-i sledeće izraze:

  a) $\frac{log_{10}100}{log_{10}10}$

  b) $\left\lfloor \frac{1+cos\frac{\pi}{4}}{1.2} \right\rfloor$

- Rešenje:
  - *log10(100)/log10(10)*
  - *floor((1+cos(Base.MathConstants.pi/4))/1.2)*
    # julia 0.2 floor((1+cos(pi/4))/1.2), bez naziva paketa

Julia - Zadatak

- **Zadatak 1.1:** Napisati u Juli-i sledeće izraze:
  - $\sqrt{3^2 + 4^2}$
  - $\left\lceil \frac{\sin(\frac{\pi}{2} - 5)}{e^5} \right\rceil$
  - $|(100 - 2^7) * 5!|$
  - *Poslednja cifra broja* $\left\lceil \tan \frac{3*\pi}{8} \right\rceil$
  - *Zapremina lopte poluprešnika 5. Formula:* $V = \frac{4}{3} r^3 \pi$

Julia - Tipovi podataka

- Julia pripada grupi dinamičkih sistema tipiziranja gde se ništa ne zna o tipovima dok program ne počne sa radom (runtime), ali poseduje i mogućnost da se eksplicitno navede da su određene vrednosti određenog tipa

- Statičko tipiziranje zahteva da je svaki tip koji figuriše u nekom izrazu poznat pre izvršavanja programa (compile time)

- Vrednosti imaju tipove, ne promenljive, promenljiva je samo naziv za vrednost

- Eksplicitno zadavanja tipa je za sada moguće samo unutar funkcija, nije moguće nad globalnim promenljivima

Julia - Tipovi podataka - Skalar

- Najčešći skalarni tipovi su:
  - Int64
  - Float64
  - Char (npr. x = 'a')
  - String (npr. x="abc")
  - Bool

- Tip podatka se može proveriti upotrebom funkcije *typeof(naziv_promenljive)*

Julia - Tipovi podataka - Stringovi

- Nad stringovima je moguće primeniti različite operacije, kao što su split, replace, join, strip, itd.

- Konkatencija (spajanje) stringova je moguće na sledeće načine:
  - Upotrebom operatora * : “abc” * “def” daje za rezultat: “abcdef”
  - Upotrebom funkcije string: string(“abc”, “def”) daje za rezultat “abcdef”
  - Upotrebom operatora $ ispred naziva promenljive: str = “abc”; println("tekst: $str");

- Upotreba prvog načina neće automatski pretvoriti vrednosti drugih tipova u tip string, za pretvaranje vrednosti bilo kog tipa u string može se koristiti funkcija string()

Julia - Tipovi podataka - Opsezi

- Opseg vrednosti može se pisati u obliku A:K:B i predstavlja opseg brojeva u intervalu [A..B] sa korakom K.
- Kada je K = 1, može se izostaviti pa se piše A:K
- Primeri:
  - *1:100, opseg brojeva od 1 do 100*
  - *1:2:100, opseg svih neparnih brojeva od 1 do 100*
  - *2:2:100, opseg svih parnih brojeva od 2 do 100*
- Opseg 1:5 NIJE isto sto i niz brojeva [1 2 3 4 5]
- Da bi se od opsega 1:5 dobio niz [1 2 3 4 5], koristi se funkcija collect()
  - *collect(1:5) = [1 2 3 4 5]*

Julia - Tipovi podataka - Nizovi i matrice

- Nizovi su n-dimenzionalne, promenljive, strukture podataka
- S obzirom da nizovi mogu da sadrze podatke razlicitih tipova oni su u sustini liste
- Niz mozemo da kreiramo upotrebom operatora uglastih zagrada: a = [] pri čemu ispred uglastih zagrada možemo da navedemo i tip elemenata niza: a = Int64[] (0-element Array{Int64,1})

Julia - Tipovi podataka - Nizovi i matrice

- **Primer 1.2**
  - *a = [1 2 3 4]*
  - *A = [1 3 5; 2 4 5]*
  - *B = collect(1:3:4) <- pretvaranje u niz*

- **Primer 1.3**
  - *A = [1 2; 3 4];*
  - *B = [1 3 5; 2 4 5];*
  - *C = [2 4; 5 6; 7 5];*
  - *AB = [A B]; #AB = [A; B] nije moguce*
  - *AC = [A; C]; #AC = [A C] nije moguce*

Julia - Tipovi podataka - Nizovi i matrice

Pristupanje elementima

- Postoje dva osnovna načina za pristupanje elementima matrice/vektora:
  - *1) Korišćenjem indeksa elementa*
    - a) Elementi se indeksiraju sa jednim brojem (indeksom), tada se elementi broje tako što se prvo prolazi kroz prvu kolonu elemenata matrice, pa zatim kroz sve ostale kolone redom,
    - b) Elementi se indeksiraju sa po jednom brojem za svaku dimenziju matrice, tako da jedan broj predstavlja indeks vrste, drugi indeks kolone.
  - *2) Hoću/neću indeksiranje upotrebom matrice logičkih vrednosti iste veličine*
  - *Operator : se koristi kako bi označio sve indekse niza.*
  - *Operator end se koristi pri indeksiranju elemenata matrice/vektora i odnosi se na indeks poslednjeg elementa u nizu (matrici, vektoru, vrsti matrice, koloni matrice).*

Julia - Tipovi podataka - Nizovi i matrice

Pristupanje elementima

Primer 1.4

A = [1 3 4; 3 4 1; 2 4 8]

- Indeksiranje jednim brojem: A[4] npr. daje za rezultat skalar 3
- Indeksiranje pomoću dva broja: A[1,3] daje za rezultat skalar 4
- Indeksiranje opsegom vrednosti: A[2:3, 2:3] daje za rezultat matricu [4 1; 4 8]
- Indeksiranjem opseg upotrebom specijalne vrednosti end A[1:end, 1:end] za rezultat vraća opet istu matricu
- Korišćenjem operatora : dobijaju se svi elementi u nekom opsegu i to odgovara opsegu 1:end
  - *A[:, 1] svi elementi prve kolone*

Julia - Tipovi podataka - Nizovi i matrice

Pristupanje elementima

Primer 1.5

A = [1 3 4; 3 4 1; 2 4 8]

- Indeksiranje pomocu logicke matrice B = [true true false; false false true; true true true] daje za rezultat vektor kolone [1; 2; 3; 4; 1; 8]

Logičku matricu B moguće je krairati od pomoću osobina elemenata matrice A

- A[A.>3] – vraća sve elemente matrice A koji su veći od 3
- A[(A.%2).==1] – vraća sve neparne elemente matrice A (A.%2 su ostaci pri deljenju elemenata matrice A sa brojem 2, a zatim te ostatke poredimo sa 1)
- A[A.!=4] – vraća sve elemente matrice A koji su različiti od 4

Julia - Tipovi podataka - Nizovi i matrice

Pristupanje elementima

- U jednom trenutku, moguće je pristupiti više elemenata niza

Primer 1.6

A = [2 4 6 8 10 12]

- A[[1 2]] = A[[2 1]] – menja prvi i drugi element niza A
- A[[1 end]] = A[[end 1]] – menja prvi i poslednji element niza A
- A[1:2:end] .= 0 - postavlja sve elmente na neparnim pozicijama na vrednost 0
- A[1:2:end] = A[2:2:end] – kopira elemente niza A na parnim pozicijama na neparne pozicije u niz A

Julia - Tipovi podataka - Nizovi i matrice

Funkcije za inicijalizaciju

- Postoji više ugrađenih funkcija koje se mogu iskoristiti u ovu svrhu i to:
  - *zeros, kreira matricu sastavljanu od nula*
  - *ones, kreira matricu sastavljanu od jedinica*
  - *Matrix{Int64}(I, N, N) – kreira jedniničnu matricu dimenzije NxN*
  - *triu, kreira gornju trougaonu matricu*
  - *tril, kreira donju trougaonu matricu*
  - *Diagonal, kreira matricu sa datom dijagonalom*
  - *Random.rand(n, m) – kreira matricu slučajnih brojeva dimenzije nxm. Generisane vrednosti su iz intervala [0, 1)*
- Velik broj ovih funkcija ima višestruko značenje i način upotrebe
  - *Diagonal([1 3 4; 3 4 1; 2 4 8]) vraća matricu sa dijagonalom 1, 4 i 8*
  - *Diagonal([1, 2, 3]) kreira matricu sa dijagonalom 1, 2, 3*
- Za detalje o upotrebi ovih funkcija konsultovati dokumentaciju
- Za neke od ovih funkcija je neophodno uvrstiti paket LinearAlgebra: using LinearAlgebra

Julia - Tipovi podataka - Nizovi i matrice

Primer 1.7. Inicijalizacija matrice. Napisati kod koji inicijalizuje matricu B veličine 4x7 koja ima sve elemente jednake 3.

```
A = ones(4,7)*3
```

Primer 1.8 Inicijalizacija matrice i pristupanje elementima matrice. Napisati kod koji inicijalizuje matricu B veličine 4x4 koja na glavnoj dijagonali ima vrednosti 1, 2, 3, 4, a sve ostale vrednosti su joj 8.

```
A = Diagonal([1, 2, 3, 4])+zeros(4,4)
A[A.==0] .= 8 # rezultat operacije nije matrica,
A             # ispisati matricu da se vidi da su promene primenjene
```

Julia - Tipovi podataka - Nizovi i matrice

Funkcije za obradu matrica

- Postoji više ugrađenih funkcija koje se mogu iskoristiti u ovu svrhu i to:
  - *sum(A) – računa sumu svih elemenata niza/matrice*
  - *maximum(A) – traži najveći broj u nizu/matrici*
  - *minimum(A) - traži najmanji broj u nizu/matrici*
  - *findall(A) – pronalazi pozicije svih nenula elemenata*
  - *reverse(A, dims = 1) – obrće matricu po vertikali (gore dole)*
  - *reverse(A, dims = 2) – obrće matricu po horizontali (levo desno)*
  - *size(A) – vraća dimenzije matrice A*
  - *size(A, 1) – vraća broj vrsta matrice A*
  - *size(A, 2) – vraća broj kolona matrice A*
  - *Statistic.mean() –vraća prosek elemenata niza/matrice*

- **Napomena:** kod većine funkcija, dodatkom parametra dims = 1 ili dims = 2, tražena vrednost se traži nad kolonama/vrstama posebno i dobija se niz vrednosti

Julia - Tipovi podataka - Nizovi i matrice

Primer 1.9. Određivanje maksimalnog elementa matrice u neparnim kolonama. Odrediti maksimalni element matrice A koji se nalazi u neparnim kolonama.

```
maxEl = maximum(A[:, 1:2:end])
```

Primer 1.10 Pozicija minimalnih elemenata matrice po parnim kolonama. Odrediti pozicije svih minimalnih elemenata u parnim kolonama matrice A

```
minEl = minimum(A[:,2:2:end])
pos = findall(A[:,2:2:end].==minEl)
pos
```

Julia - Tipovi podataka - Nizovi i matrice

Primer 1.11 Rad sa matricama i pristupanje elementima matrice. Napisati kod koji iz zadate matrice A veličine 5x5 uzima elemente koji se nalaze na sporednoj dijagonali i veći su od 7

```
sd = Diagonal(reverse(A, dims=2))
rez = sd[sd.>7]
```

Primer 1.12 Pristupanje elementima matrice. Napisati kod koji generišu matricu A veličine 6x6, koja ima sve jedinice, osim u uglovima gde se nalaze dvojke.

```
A = ones(6,6)
A[[1, end], [1, end]] .= 2 # rezultat operacije nije matrica,
A           # ispisati matricu da se vidi da su promene primenjene
```

Julia - Tipovi podataka - Nizovi i matrice

Primer 1.13 Neka je matricom A predstavljen pregled ocena studenata, gde svaka vrsta predstavlja različitog studenta. U prvoj koloni se nalaze brojevi indeksa studenata, a u ostalim ocene studenata po predmetima (svaki predmet jedna kolona).

1) Odrediti ukupan broj studenata i broj predmeta.

2) Odrediti koliko je studenata ispunilo uslov da upiše višu godinu (uslov je najviše dva nepoložena ispita).

3) Odrediti redni broj najbolje ocenjenog predmeta i prosečnu ocenu na tom predmetu (ako ih ima više sa istom prosečnom ocenom – uzimamo redne brojeve svih).

4) Odrediti redni broj predmeta sa najlošijom prolaznošću (ako ih ima više sa istom prolaznošću - najlošijom, uzimamo redne brojeve svih).

5) Odrediti indeks studenta sa najboljim prosekom (ako ih ima vise sa istim prosekom – najboljim, vratiti u vektoru indekse svih).

Julia - Tipovi podataka - Nizovi i matrice

Primer matrice A:

123	6	9	5	8	7	9	5
234	8	9	7	8	7	10	10
345	8	7	6	5	8	7	7
456	5	9	6	5	7	8	5
567	6	9	9	7	9	7	8
789	9	7	9	8	7	9	10

Julia - Tipovi podataka - Nizovi i matrice

Rešenja primera 1.13:

```
A = [123 6 9 5 8 7 9 5;234 8 9 7 8 7 10 10;345 8 7 6 5 8 7 7;456 5 9 6 5 7 8 5;567 6
9 9 7 9 7 8;789 9 7 9 8 7 9 10]
(brojRedova, brojKolona) = size(A)
brojStudenata = brojRedova
brojPredmeta = brojKolona - 1
matricaOcena = A[:,2:end]
matricaNepolozenihIspita = matricaOcena.==5
vektorIspunjenUslov = sum(matricaNepolozenihIspita,dims=2).<=2 #0.2: sum(A, 1)
brojIspunjenUslov = sum(vektorIspunjenUslov)
vektorProsecnihOcenaNaPredmetu = Statistics.mean(matricaOcena, dims=1)
prosecnaOcenaNajboljeOcenjenogPredmeta = maximum(vektorProsecnihOcenaNaPredmetu)
redniBrojNajboljeOcenjenogPredmeta = findall(vektorProsecnihOcenaNaPredmetu .==
prosecnaOcenaNajboljeOcenjenogPredmeta)
vektorNeprolaznostiPoPredmetu = sum(matricaNepolozenihIspita, dims=1)
redniBrojPredmetaSaNajlosijomProlaznoscu =
findall(vektorNeprolaznostiPoPredmetu.==maximum(vektorNeprolaznostiPoPredmetu))
vektorProsekaStudenata = Statistics.mean(matricaOcena, dims=2) #0.2: samo mean(A, 2)
```

Julia - Tipovi podataka - Nizovi i matrice

Rešenja primera 1.13:

```
najboljiProsek = maximum(vektorProsekaStudenata)
redniBrojeviStudenataSaNajboljomOcenom = findall(vektorProsekaStudenata .==
najboljiProsek)
indeksiStudenataSaNajboljomOcenom = A[redniBrojeviStudenataSaNajboljomOcenom]
```

Julia - Tipovi podataka - Nizovi i matrice

Zadaci za samostalnu vežbu:

1. Formirati matricu 4x4 koja iznad glavne dijagonale ima vrednosti 5, ispod nje 2, a na glavnoj dijagonali vrednosti 4.
2. Za zadatu matricu A veličine 10x10, čije su vrednosti slučajno generisane, napisati program koji određuje sve elemente iznad sporedne dijagonale (izdvaja u poseban vektor).
3. Neka je zadata proizvoljna matrica A. Odrediti redni broj vrste sa maksimalnim zbirom elemenata.
4. Neka je zadata proizvoljna matrica B. Napisati kod u Julia-i koji određuje logičku vrednost da li je proizvod svih elemenata u parnim redovima veći od sume svih elemenata u neparnim kolonama.
5. Neka je dan sistem od 3 linearne jednačine sa 3 nepoznate. Napisati lkod koji rešava dati sistem