

Predavanje 3 - Strukture podataka

Tatjana Tomović

Institut za matematiku i informatiku
Prirodno-matematički fakultet
Univerzitet u Kragujevcu

Kragujevac, 2014.

Pregled predavanja

1 Matrice

- Operacije sa matricama

2 Nizovi karaktera

- Komparacija stringova
- Pretraga stringa i zamena podstringova

3 Nizovi ćelija

Kreiranje jednostavnih matrica

- Matrice se najjednostavnije kreiraju koristeći operator konstrukcije `[ ]`.

Kreiranje jednostavnih matrica

- Matrice se najjednostavnije kreiraju koristeći operator konstrukcije `[ ]`.
- Između elemenata se može navoditi zarez ili razmak.

Kreiranje jednostavnih matrica

- Matrice se najjednostavnije kreiraju koristeći operator konstrukcije `[ ]`.
- Između elemenata se može navoditi zarez ili razmak.
- Između dve vrste se navodi znak `;`.

Kreiranje jednostavnih matrica

- Matrice se najjednostavnije kreiraju koristeći operator konstrukcije [].
- Između elemenata se može navoditi zarez ili razmak.
- Između dve vrste se navodi znak ;.

Ulaz:

```
>> vrsta = [1, 2, 3, 4]
```

```
>> vrsta = [1 2 3 4]
```

```
>> I3 = [1 0 0; 0, 1, 0; 0 0, 1];
```

```
>> a = ' dobar dan';
```

```
>> whos vrsta
```

```
>> whos vrsta
```

Kreiranje jednostavnih matrica

- Da bismo pristupali elementima matrice, koristimo operator (,).

Kreiranje jednostavnih matrica

- Da bismo pristupali elementima matrice, koristimo operator $(,)$.
- Indeks vrste (kolone), počinje od jedinice.

Kreiranje jednostavnih matrica

- Da bismo pristupali elementima matrice, koristimo operator (,).
- Indeks vrste (kolone), počinje od jedinice.

```
>> a = [1 2, 3 4];  
>> a(0,0)          >> a(1,3)  
>> a = 'dobar dan';  
>> a(1,1)  
>> a = []          >> whos a
```

Kreiranje jednostavnih matrica

- Da bismo pristupali elementima matrice, koristimo operator (,).
- Indeks vrste (kolone), počinje od jedinice.

```
>> a = [1 2, 3 4];  
>> a(0,0)          >> a(1,3)  
>> a = 'dobar dan';  
>> a(1,1)  
>> a = []          >> whos a
```

Naredba `a = []` kreira praznu matricu `a` tipa 0×0 .

Kreiranje jednostavnih matrica

<i>ones</i>	kreira matricu zadatog tipa ispunjenu jedinicama
<i>zeros</i>	kreira matricu zadatog tipa ispunjenu nulama
<i>eye</i>	kreira jediničnu matricu zadatog tipa
<i>diag</i>	kreira dijagonalnu matricu od zadate matrice tipa $1 \times n$ ili $n \times 1$
<i>magic</i>	kreira kvadratnu matricu čije vrste, kolone i dijagonale imaju jednake zbirove
<i>rand</i>	kreira matricu čiji su elementi slučajni brojevi sa uniformnom raspodelom
<i>randn</i>	isto što i <i>rand</i> , ali su elementi matrice sa normalnom distribucijom
<i>randperm</i>	kreira matricu tipa $1 \times n$ koja sadrži slučajnu permutaciju zadatih celobrojnih vrednosti

Kreiranje jednostavnih matrica

```
>> magic(5)  
>> a = rand(3,5);  
>> a = diag([1 2,3])  
>> b = diag([1 2 3],1)  
>> b = diag([1 2],-2)
```

Kreiranje jednostavnih matrica

```
>> magic(5)
>> a = rand(3,5);
>> a = diag([1 2,3])
>> b = diag([1 2 3],1)
>> b = diag([1 2],-2)
```

Vidimo da je moguće vektor kojim se definiše matrica postaviti i na sporednim dijagonalama. Na primer, u formiranju matrice *b* jedinica, drugi argument pri pozivu funkcije *diag*, znači da vektor treba postaviti na prvoj dijagonali iznad glavne dijagonale.

Kreiranje jednostavnih matrica

```
>> magic(5)
>> a = rand(3, 5);
>> a = diag([1 2, 3])
>> b = diag([1 2 3], 1)
>> b = diag([1 2], -2)
```

Vidimo da je moguće vektor kojim se definiše matrica postaviti i na sporednim dijagonalama. Na primer, u formiranju matrice *b* jedinica, drugi argument pri pozivu funkcije *diag*, znači da vektor treba postaviti na prvoj dijagonali iznad glavne dijagonale. Ako hoćemo da postavimo vektor ispod koristimo bismo negativne cele brojeve.

Konkatenacija matrica

- Konkatenacija se izvodi koristeći operator $[\]$.

Konkatenacija matrica

- Konkatenacija se izvodi koristeći operator $[]$.
- Ako su a i b matrice, koje zadovoljavaju odgovarajuća ograničenja, izraz $[a \ b]$ predstavlja konkatenaciju matrica horizontalno, dok izraz $[a; b]$ predstavlja konkatenaciju matrica vertikalno.

Konkatenacija matrica

- Konkatenacija se izvodi koristeći operator `[ ]`.
- Ako su a i b matrice, koje zadovoljavaju odgovarajuća ograničenja, izraz `[a b]` predstavlja konkatenaciju matrica horizontalno, dok izraz `[a; b]` predstavlja konkatenaciju matrica vertikalno.

```
>> a = ones(1, 3);    >> b = rand(2, 3);  
>> c = [a; b]  
>> a = ones(2, 2);    >> b = rand(2, 3);  
>> c = [a b]
```

Konkatenacija matrica

<i>cat</i>	konkatenira matrice po zadatoj dimenziji
<i>horzcat</i>	konkatenira matrice horizontalno
<i>vertcat</i>	konkatenira matrice vertikalno
<i>repmat</i>	konstruise novu matricu replikacijom i tiling-om zadatih matrica
<i>blkdiag</i>	$1 \times n$ ili $n \times 1$
<i>magic</i>	kreira blok dijagonalnu matricu od postojećih matrica

```
>> a = [1, 2; 3, 4; 5, 6];    >> b = repmat(a, 2, 3)
>> a = [1, 2, 3];           >> b = ones(2);
>> c = [1; 2];               >> d = blkdiag(a, b, c)
```

Kreiranje aritmetičkih nizova

- Operator : deluje tako što formira aritmetički niz sa korakom 1 i početnim elementom koji je levi operand. Aritmetički niz staje kad se dobije element koji je veći od desnog operanda, i taj, prvi veći element, nije uključen u rezultujuću matricu.

Kreiranje aritmetičkih nizova

- Operator : deluje tako što formira aritmetički niz sa korakom 1 i početnim elementom koji je levi operand. Aritmetički niz staje kad se dobije element koji je veći od desnog operanda, i taj, prvi veći element, nije uključen u rezultujuću matricu.
- Korak sa kojim se formira aritmetički niz se zadaje između početne vrednosti i vrednosti koja predstavlja gornju granicu.

Kreiranje aritmetičkih nizova

- Operator : deluje tako što formira aritmetički niz sa korakom 1 i početnim elementom koji je levi operand. Aritmetički niz staje kad se dobije element koji je veći od desnog operanda, i taj, prvi veći element, nije uključen u rezultujuću matricu.
- Korak sa kojim se formira aritmetički niz se zadaje između početne vrednosti i vrednosti koja predstavlja gornju granicu.

```
>> a = [1 2 3 4];    >> b = 1 : 4  
>> a = -2.5 : 2.5    >> a = 1 : 1.2  
>> a = 1 : 2 : 10.1
```

Kreiranje matrica sa različitim tipovima podataka

- Moguće je da prilikom kreiranja matrice različiti elementi matrice pripadaju različitim tipovima (klasama) podataka.

Kreiranje matrica sa različitim tipovima podataka

- Moguće je da prilikom kreiranja matrice različiti elementi matrice pripadaju različitim tipovima (klasama) podataka.
- Prilikom kreiranja matrica sa različitim tipovima podataka vrši se konverzija tipova svih elemenata matrice tako da u kreiranoj matrici svi elementi imaju isti tip.

Kreiranje matrica sa različitim tipovima podataka

- Moguće je da prilikom kreiranja matrice različiti elementi matrice pripadaju različitim tipovima (klasama) podataka.
- Prilikom kreiranja matrica sa različitim tipovima podataka vrši se konverzija tipova svih elemenata matrice tako da u kreiranoj matrici svi elementi imaju isti tip.

```
>> a = [68 single(1) 2.3 'a']      >> whos a  
>> a = [single(1.2) 1.3 1e + 300]
```

Adresiranje elemenata matrica

- Matrice su u memoriji smeštene linearno, po kolonama, pa elementima matrice možemo pristupiti kao elementima niza, dakle, koristeći jedan indeks.

Adresiranje elemenata matrica

- Matrice su u memoriji smeštene linearno, po kolonama, pa elementima matrice možemo pristupati kao elementima niza, dakle, koristeći jedan indeks.
- Postoje funkcije Matlaba koje se mogu koristiti za preračunavanja položaja u memoriji i položaja elementa u matrici. To su funkcije *sub2ind* koja od pozicije elementa u matrici preračunava linearni indeks, i *ind2sub* koja na osnovu linearnog indeksa izračunava poziciju elementa u matrici.

Adresiranje elemenata matrica

- Matrice su u memoriji smeštene linearno, po kolonama, pa elementima matrice možemo pristupiti kao elementima niza, dakle, koristeći jedan indeks.
- Postoje funkcije Matlaba koje se mogu koristiti za preračunavanja položaja u memoriji i položaja elementa u matrici. To su funkcije *sub2ind* koja od pozicije elementa u matrici preračunava linearni indeks, i *ind2sub* koja na osnovu linearnog indeksa izračunava poziciju elementa u matrici.

```
>> a = [2, 3, 4; 5 6 7; 8, 9, 10];    >> a(4)
>> a(sub2ind(size(a), 2, 2))
>> a(ind2sub(size(a), 5))
```

Adresiranje elemenata matrica

- Izraz $b(1 : 4, 6)$ formira matricu tipa 4×1 , koja za elemente ima prvi, drugi, treći i četvrti element šeste kolone matrice b .

Adresiranje elemenata matrica

- Izraz $b(1 : 4, 6)$ formira matricu tipa 4×1 , koja za elemente ima prvi, drugi, treći i četvrti element šeste kolone matrice b .
- Izraz $a(2, 1 : 2)$ formira matricu tipa 1×2 koja za elemente ima elemente druge vrste matrice a , na pozicijama 1 i 2.

Adresiranje elemenata matrica

- Izraz $b(1 : 4, 6)$ formira matricu tipa 4×1 , koja za elemente ima prvi, drugi, treći i četvrti element šeste kolone matrice b .
- Izraz $a(2, 1 : 2)$ formira matricu tipa 1×2 koja za elemente ima elemente druge vrste matrice a , na pozicijama 1 i 2.
- Koristeći linearno indeksiranje možemo pristupiti pojedinačnim elementima matrice koji nisu pripadnici jedne vrste ili kolone.

Adresiranje elemenata matrica

- Izraz $b(1 : 4, 6)$ formira matricu tipa 4×1 , koja za elemente ima prvi, drugi, treći i četvrti element šeste kolone matrice b .
- Izraz $a(2, 1 : 2)$ formira matricu tipa 1×2 koja za elemente ima elemente druge vrste matrice a , na pozicijama 1 i 2.
- Koristeći linearno indeksiranje možemo pristupiti pojedinačnim elementima matrice koji nisu pripadnici jedne vrste ili kolone.
- Jedna od primena ključne reči *end* je da označi poslednju vrstu poziciji vrste ili poslednju kolonu.

Adresiranje elemenata matrica

- Izraz $b(1 : 4, 6)$ formira matricu tipa 4×1 , koja za elemente ima prvi, drugi, treći i četvrti element šeste kolone matrice b .
- Izraz $a(2, 1 : 2)$ formira matricu tipa 1×2 koja za elemente ima elemente druge vrste matrice a , na pozicijama 1 i 2.
- Koristeći linearno indeksiranje možemo pristupiti pojedinačnim elementima matrice koji nisu pripadnici jedne vrste ili kolone.
- Jedna od primena ključne reči *end* je da označi poslednju vrstu poziciji vrste ili poslednju kolonu.

```
>> a = [2, 3, 4; 5 6 7; 8, 9, 10];    >> a(2 : 3, 3)
>> b = ones(4, 5);
>> b(1 : 3 : 20) = 2;                >> b
>> b(1 : 3 : end)                    >> b(1 : 2, end)
```

Adresiranje elemenata matrica

- Operator : može se koristiti da označi sve elemente jedne kolone ili vrste.

Adresiranje elemenata matrica

- Operator : može se koristiti da označi sve elemente jedne kolone ili vrste.
- Moguće je koristiti matricu kao indeks.

Adresiranje elemenata matrica

- Operator : može se koristiti da označi sve elemente jedne kolone ili vrste.
- Moguće je koristiti matricu kao indeks.
- Indeksiranje pomoću matrice koristi linearno indeksiranje, dakle, svaki element matrice koja se koristi kao indeks se tumači kao linearni indeks elementa matrice koja se adresira. Rezultujuća matrica naravno ima tip matrice kojom se adresira, a ima elemente matrice koja se adresira.

Adresiranje elemenata matrica

- Operator : može se koristiti da označi sve elemente jedne kolone ili vrste.
- Moguće je koristiti matricu kao indeks.
- Indeksiranje pomoću matrice koristi linearno indeksiranje, dakle, svaki element matrice koja se koristi kao indeks se tumači kao linearni indeks elementa matrice koja se adresira. Rezultujuća matrica naravno ima tip matrice kojom se adresira, a ima elemente matrice koja se adresira.

```
>> a(:,2)           >> a(end,:)
>> a(9 * ones(2,10))
```

Adresiranje elemenata matrica

- Elementi matrice se mogu adresirati matricom čiji su elementi tipa *logical*. Pri tome, samo pozicije koje u matrici koja adresira imaju vrednost *true*, bivaju promenjene naredbom dodele.

Adresiranje elemenata matrica

- Elementi matrice se mogu adresirati matricom čiji su elementi tipa *logical*. Pri tome, samo pozicije koje u matrici koja adresira imaju vrednost *true*, bivaju promenjene naredbom dodele.

```
>> a = rand(5)    >> b = a > 0.5  
>> a(b) = 0       >> a(b)
```

Adresiranje elemenata matrica

- Elementi matrice se mogu adresirati matricom čiji su elementi tipa *logical*. Pri tome, samo pozicije koje u matrici koja adresira imaju vrednost *true*, bivaju promenjene naredbom dodele.

```
>> a = rand(5)    >> b = a > 0.5  
>> a(b) = 0       >> a(b)
```

Matrica $a(b)$ sadrži samo one elemente koji se nalaze na onim mestima na kojima matrica koja adresira ima logičke vrednosti *true*.

Funkcije za prikupljanje informacija o matricama

<i>length</i>	vraća maksimum broja elemenata po svim dimenzijama
<i>ndims</i>	vraća broj dimenzija matrice
<i>numel</i>	vraća broj elemenata matrice
<i>size</i>	vraća tip matrice
<i>isempty</i>	daje informaciju da li je broj elemenata u nekoj dimenziji jednak nula
<i>isscalar</i>	daje informaciju da li je argument matrica tipa 1×1
<i>issparse</i>	daje informaciju da li je argument retka matrica
<i>isvector</i>	daje informaciju da li je argument matrica tipa $1 \times n$ ili $n \times 1$

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.
- Jedan način za povećanje broja elemenata je dodela vrednosti elementu matrice koji u trenutku dodele ne postoji.

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.
- Jedan način za povećanje broja elemenata je dodela vrednosti elementu matrice koji u trenutku dodele ne postoji.
- Ostali elementi matrice koji su neophodni da bi matrica zadržala strukturu matrice se inicijalizuju na vrednost 0.

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.
- Jedan način za povećanje broja elemenata je dodela vrednosti elementu matrice koji u trenutku dodele ne postoji.
- Ostali elementi matrice koji su neophodni da bi matrica zadržala strukturu matrice se inicijalizuju na vrednost 0.
- Moguće je i smanjiti broj elemenata u bilo kojoj dimenziji matrice, tako sto se izaberu vrste (kolone) matrice kojima treba dodeliti praznu matricu.

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.
- Jedan način za povećanje broja elemenata je dodela vrednosti elementu matrice koji u trenutku dodele ne postoji.
- Ostali elementi matrice koji su neophodni da bi matrica zadržala strukturu matrice se inicijalizuju na vrednost 0.
- Moguće je i smanjiti broj elemenata u bilo kojoj dimenziji matrice, tako sto se izaberu vrste (kolone) matrice kojima treba dodeliti praznu matricu.

Promena tipa matrice

- Svim matricama u Matlabu moguće je promeniti tip.
- Jedan način za povećanje broja elemenata je dodela vrednosti elementu matrice koji u trenutku dodele ne postoji.
- Ostali elementi matrice koji su neophodni da bi matrica zadržala strukturu matrice se inicijalizuju na vrednost 0.
- Moguće je i smanjiti broj elemenata u bilo kojoj dimenziji matrice, tako sto se izaberu vrste (kolone) matrice kojima treba dodeliti praznu matricu.

```
>> a = ones(2,3);    >> a(2,7) = 2  
>> a = ones(2,3);  
>> a(:,4:7) = 2;     >> a(3:4,:) = 3  
>> a(:,2) = [];      >> a(4,:) = []
```

Promena tipa matrice

<i>reshape</i>	menja tip matrice
<i>rot90</i>	rotira matricu za 90° , u smislu pravougaone šeme brojeva
<i>fliplr</i>	okreće matricu oko vertikalne ose, u smislu pravougaone šeme brojeva
<i>isempty</i>	daje informaciju da li je broj elemenata u nekoj dimenziji jednak nula
<i>flipud</i>	okreće matricu oko horizontalne ose, u smislu pravougaone šeme brojeva
<i>transpose</i>	transponuje matricu
<i>ctranspose</i>	transponuje matricu i konjuguje njene elemente

Promena tipa matrice

<i>reshape</i>	menja tip matrice
<i>rot90</i>	rotira matricu za 90° , u smislu pravougaone šeme brojeva
<i>fliplr</i>	okreće matricu oko vertikalne ose, u smislu pravougaone šeme brojeva
<i>isempty</i>	daje informaciju da li je broj elemenata u nekoj dimenziji jednak nula
<i>flipud</i>	okreće matricu oko horizontalne ose, u smislu pravougaone šeme brojeva
<i>transpose</i>	transponuje matricu
<i>ctranspose</i>	transponuje matricu i konjuguje njene elemente

Funkcija *reshape* formira novu matricu, koja ima isti broj elemenata kao matrica u argumentu. Dakle, ako je matrica u argumentu imala tip $n \times m$, i rezultujuća matrica ima tip $p \times q$, onda mora biti ispunjen uslov $nm = pq$.

Promena tipa matrice

Zadatak: za svaku od prethodno datih funkcija pronaci objasnjenje u help-u i primeniti je na proizvoljnom primeru.

Aritmetičke operacije

$+$	sabiranje	$-$	oduzimanje
$.*$	množenje	$./$	desno deljenje
$.\backslash$	levo deljenje	$+$	unarni operator $+$
$-$	unarni operator $-$	$:$	više namena
$.^$	stepenovanje	$.'$	transponovanje
$'$	transponovanje i konjugacija	$*$	matrično množenje
$/$	desno matrično deljenje	$\backslash$	matrično levo deljenje
$^$	matrično stepenovanje		
<i>sum</i>	sumira sve elemente matrice	<i>prod</i>	množi sve elemente matrice

Aritmetičke operacije

$+$	sabiranje	$-$	oduzimanje
$.*$	množenje	$./$	desno deljenje
$.\backslash$	levo deljenje	$+$	unarni operator $+$
$-$	unarni operator $-$	$:$	više namena
$.^{\wedge}$	stepenovanje	$.'$	transponovanje
$'$	transponovanje i konjugacija	$*$	matrično množenje
$/$	desno matrično deljenje	$\backslash$	matrično levo deljenje
$^{\wedge}$	matrično stepenovanje		
<i>sum</i>	sumira sve elemente matrice	<i>prod</i>	množi sve elemente matrice

Svi operatori koji počinju karakterom `.` označavaju da se operacije izvode nad elementima, a ne nad matricama shvaćenim kao jedinstvenim algebarskim objektima. Odstupa samo `.'`.

Aritmetičke operacije

Data su matrice

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -9 \end{bmatrix}$$

1. Odrediti $3 + A$.
2. Pomnožiti elemente matrica A i B koji su na istim pozicijama.
3. Pomnožiti svaki element matrice A sa 2.
4. Promeniti znak svim elementima matrice A .
5. Odrediti $A \cdot B$.
6. Odrediti inverznu matricu matrice A .

Aritmetičke operacije

Data su matrice

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -9 \end{bmatrix}$$

1. Odrediti $3 + A$.
2. Pomnožiti elemente matrica A i B koji su na istim pozicijama.
3. Pomnožiti svaki element matrice A sa 2.
4. Promeniti znak svim elementima matrice A .
5. Odrediti $A \cdot B$.
6. Odrediti inverznu matricu matrice A .

Operatori koji deluju u matričnom smislu ne mogu biti primenjeni nad matricama čiji su elementi tipa *integer*.

Relacioni operatori

$<$	manje	$>$	veće
$<=$	manje ili jednako	$>=$	veće ili jednako
$==$	jednako	$\sim$	različito

Relacioni operatori

$<$	manje	$>$	veće
$<=$	manje ili jednako	$>=$	veće ili jednako
$==$	jednako	$\sim$	različito

Relacioni operatori proizvode kao rezultat uvek vrednosti tipa *logical*. Relacioni operatori uvek dejstvuju na odnosne elemente, i kao rezultat proizvode matrice odgovarajućeg tipa.

Relacioni operatori

<	manje	>	veće
<=	manje ili jednako	>=	veće ili jednako
==	jednako	~=	različito

Relacioni operatori proizvode kao rezultat uvek vrednosti tipa *logical*. Relacioni operatori uvek dejstvuju na odnosne elemente, i kao rezultat proizvode matrice odgovarajućeg tipa.

```
>> a = [1 2 3; 4 5 6];    >> b = 2 * ones(2, 3);  
>> a >= b                >> c = a < b
```

Logički operatori

$\&$	konjukcija	$ $	disjunkcija
$\sim$	negacija	<i>not</i>	negacija
<i>and</i>	konjukcija	<i>or</i>	disjunkcija
<i>xor</i>	eksluzivna disjunkcija		
<i>and</i>	konjukcija svih elemenata vektora		
<i>any</i>	disjunkcija svih elemenata vektora		

Logički operatori

$\&$	konjukcija	$ $	disjunkcija
$\sim$	negacija	<i>not</i>	negacija
<i>and</i>	konjukcija	<i>or</i>	disjunkcija
<i>xor</i>	eksluzivna disjunkcija		
<i>and</i>	konjukcija svih elemenata vektora		
<i>any</i>	disjunkcija svih elemenata vektora		

Operatori deluju nad odnosnim elementima, pa matrice moraju biti istog tipa. Izuzetak su skalarne matrice koje mogu biti kombinovane sa matricom bilo kog tipa.

Logički operatori

$\&$	konjukcija	$ $	disjunkcija
$\sim$	negacija	<i>not</i>	negacija
<i>and</i>	konjukcija	<i>or</i>	disjunkcija
<i>xor</i>	eksluzivna disjunkcija		
<i>and</i>	konjukcija svih elemenata vektora		
<i>any</i>	disjunkcija svih elemenata vektora		

Operatori deluju nad odnosnim elementima, pa matrice moraju biti istog tipa. Izuzetak su skalarne matrice koje mogu biti kombinovane sa matricom bilo kog tipa.

```
>> a = 2;    >> b = [1 0 2; 0 0 -1; 0 -2 -3];  
>> a&b      >> and(a, b)
```

Logički operatori

Kad se *and* i *any* primenjuju nad matricama, ove funkcije deluju nad kolonama matrica i povratna vrednost su matrice vrste.

Logički operatori

Kad se *and* i *any* primenjuju nad matricama, ove funkcije deluju nad kolonama matrica i povratna vrednost su matrice vrste.

```
>> a = uint8([1 2 3 0]); >> any(a)
>> all(a)b              >> any(a')
>> all(a')
```

Prioritet operacija

Prioriteti operacija u Matlabu poređani od najvišeg ka najnižem.

Prioritet operacija

Prioriteti operacija u Matlabu poređani od najvišeg ka najnižem.

1. `()`
2. `./`, `.^`, `'`, `^`
3. unarni `+`, unarni `-`, `~`
4. `.*`, `./`, `.\`, `*`, `/`, `\`
5. `+`, `-`
6. `:`
7. `<`, `<=`, `>`, `>=`, `==`, `~=`
8. `&`
9. `|`
10. `&&`
11. `||`

Sortiranje elemenata matrice

<i>circshift</i>	cirkularno pomera elemente matrice
<i>sort</i>	sortira elemente niza u rastući ili opadajući poredak
<i>sortrows</i>	sortira vrste matrice u rastućem poretku
<i>issorted</i>	daje informaciju da li su elementi matrice sortirani
<i>min</i>	vraća vrednosti minimalnih elemenata po kolonama i njihove indekse
<i>max</i>	vraća vrednosti maksimalnih elemenata po kolonama i njihove indekse

Sortiranje elemenata matrice

<i>circshift</i>	cirkularno pomera elemente matrice
<i>sort</i>	sortira elemente niza u rastući ili opadajući poredak
<i>sortrows</i>	sortira vrste matrice u rastućem poretku
<i>issorted</i>	daje informaciju da li su elementi matrice sortirani
<i>min</i>	vraća vrednosti minimalnih elemenata po kolonama i njihove indekse
<i>max</i>	vraća vrednosti maksimalnih elemenata po kolonama i njihove indekse

Funkcija *circshift* razmešta vrste i kolone matrice tako što ih cirkularno pomera na levo ili na desno, kad su u pitanju kolone, ili nadole i nagore, kad su u pitanju vrste. Drugi argument u pozivu funkcije je matrica tipa 1×2 , čiji prvi element ukazuje na broj pozicija za koji treba premestiti vrste, a drugi element je broj pozicija za koji treba razmestiti kolone.

Sortiranje elemenata matrice

```
>> a = [1 2 3 4; 5 6 7 8; 9, 10, 11, 12];  
>> circshift(a, [-1, 2])  
>> a = rand(4, 5) * 9  
>> sort(a, 1)  
>> a = sort(rand(2, 3), 1);  
>> a = sort(rand(2, 3), 1, 'descend');  
                                     >> sort(a, 2)  
                                     >> issorted(a(:, 2))  
                                     >> issorted(a(:, 2))
```

Sortiranje elemenata matrice

```
>> a = [1 2 3 4; 5 6 7 8; 9, 10, 11, 12];  
>> circshift(a, [-1, 2])  
>> a = rand(4, 5) * 9  
>> sort(a, 1)                                     >> sort(a, 2)  
>> a = sort(rand(2, 3), 1);                        >> issorted(a(:, 2))  
>> a = sort(rand(2, 3), 1, 'descend');             >> issorted(a(:, 2))
```

Ako je drugi argument u pozivu funkcije 1 sortiranje se vrši po prvoj dimenziji, dakle, sortira se u kolonama, ako je drugi argument 2 sortiranje se vrši po drugoj dimenziji, dakle, u vrstama. Prilikom sortiranja u kolonama, vrste matrice nisu očuvane, dok, kad se sortira u vrstama, kolone matrice nisu očuvane. Funkcija *sort* sortira matricu u rastući poredak. Da bismo sortirali u opadajućem poretку postupamo na sledeći način *sort(a, 2, 'descend')*.

Sortiranje elemenata matrice

```
>> a = [1 2 3 4; 5 6 7 8; 9, 10, 11, 12];  
>> issorted(a,'rows')  
>> issorted(flipud(a),'rows')  
>> a = rand(4,5) * 9
```

`>> sortrows(a,3)`

Sortiranje elemenata matrice

```
>> a = [1 2 3 4; 5 6 7 8; 9, 10, 11, 12];  
>> issorted(a,'rows')  
>> issorted(flipud(a),'rows')  
>> a = rand(4,5) * 9                                >> sortrows(a,3)
```

Vidimo da su vrste očuvane, a sortiranje je izvedeno u trećoj koloni.

Sortiranje elemenata matrice

```
>> a = [1 2 3 4; 5 6 7 8; 9, 10, 11, 12];  
>> issorted(a,'rows')  
>> issorted(flipud(a),'rows')  
>> a = rand(4,5) * 9                                >> sortrows(a,3)
```

Vidimo da su vrste očuvane, a sortiranje je izvedeno u trećoj koloni. Funkcija *min* vraća vektor tipa $1 \times n$, koji sadrži minimalne elemente po kolonama argumenta. Ako funkcija *min* ima dva izlazna argumenta, onda je prvi izlazni argument vektor minimalnih elemenata po kolonama, a drugi izlazni argument je vektor pozicija minimalnih elemenata u kolonama.

Operacije nad dijagonalnim matricama

<i>blkdiag</i>	konstruiše dijagonalnu matricu od niza matrica
<i>diag</i>	konstruiše dijagonalnu matricu od zadatog vektora
<i>trace</i>	izračunava sumu elemenata na glavnoj dijagonali
<i>tril</i>	vraća donji trougaonu matricu zadate matrice
<i>triu</i>	vraća gornju trougaonu matricu zadate matrice

Operacije nad dijagonalnim matricama

<i>blkdiag</i>	konstruiše dijagonalnu matricu od niza matrica
<i>diag</i>	konstruiše dijagonalnu matricu od zadatog vektora
<i>trace</i>	izračunava sumu elemenata na glavnoj dijagonali
<i>tril</i>	vraća donji trougaonu matricu zadate matrice
<i>triu</i>	vraća gornju trougaonu matricu zadate matrice

Kad je funkcija *tril* navedena bez drugog argumenta vraća donji trougao. Ako je navedena sa drugima argumentom, vraća sve dijagonale počev od elementa dole levo do dijagonale kod koje je razlika indeksa vrste i kolone jednaka drugom argumentu u pozivu funkcije. Slično je i sa funkcijom *triu*. S tim, što funkcija vraća gornji trougao i počinje od elementa gore desno i završava sa dijagonalom gde je razlika indeksa vrste i kolone jednaka drugom argumentu u pozivu funkcije.

Pune i retke matrice

<i>full</i>	konvertuje matricu iz retkog oblika skladištenja u pun oblik
<i>issprase</i>	određuje da li je matrica retka ili nije
<i>nnz</i>	vraća broj elemenata matrice koji nisu jednaki nula
<i>nonzeros</i>	vraća sve elemente matrice koji nemaju vrednost nula
<i>nzmax</i>	vraća količinu memorije koja je zauzeta od strane elemenata čija vrednost nije jednaka nula
<i>spalloc</i>	alocira memoriju za skladištenje retke matrice
<i>sparse</i>	kreira retku matricu ili konvertuje punu matricu u retku
<i>speye</i>	kreira jediničnu matricu u retkom obliku
<i>sprand</i>	kreira retku matricu čiji su elementi slučajni brojevi uniformno raspodeljeni

Višedimenzionalni nizovi

Višedimenzionalni nizovi su ekstenzija matrica u Matlabu.

Višedimenzionalni nizovi

Višedimenzionalni nizovi su ekstenzija matrica u Matlabu. Na primer, pozicija elementa trodimenzionalnog niza je određena sa tri indeksa: prvi je indeks vrste, drugi indeks kolone a treći je indeks stranice.

Višedimenzionalni nizovi

Višedimenzionalni nizovi su ekstenzija matrica u Matlabu.

Na primer, pozicija elementa trodimenzionalnog niza je određena sa tri indeksa: prvi je indeks vrste, drugi indeks kolone a treci je indeks stranice.

Trodimenzionalni nizove možemo shvatiti kao niz matrica koje su organizovane u stranice.

Nizovi karaktera

- Nizovi karaktera (podaci tipa *string*) su matrice tipa $1 \times n$.

Nizovi karaktera

- Nizovi karaktera (podaci tipa *string*) su matrice tipa $1 \times n$.
- Moguće je formirati i matrice drugih tipova.

Nizovi karaktera

- Nizovi karaktera (podaci tipa *string*) su matrice tipa $1 \times n$.
- Moguće je formirati i matrice drugih tipova.
- Prilikom kreiranja matrica koristeći stringove kao vrste svi moraju biti iste dužine, dakle, sa istim brojem karaktera.

Nizovi karaktera

- Nizovi karaktera (podaci tipa *string*) su matrice tipa $1 \times n$.
- Moguće je formirati i matrice drugih tipova.
- Prilikom kreiranja matrica koristeći stringove kao vrste svi moraju biti iste dužine, dakle, sa istim brojem karaktera.
- Ovo ograničenje nije prirodno i moguće ga je prevazići koristeći dodavanje karaktera belina (blank) na kraj svih stringova čija je dužina manja od dužine stringa čija je dužina maksimalna.

Nizovi karaktera

- Nizovi karaktera (podaci tipa *string*) su matrice tipa $1 \times n$.
- Moguće je formirati i matrice drugih tipova.
- Prilikom kreiranja matrica koristeći stringove kao vrste svi moraju biti iste dužine, dakle, sa istim brojem karaktera.
- Ovo ograničenje nije prirodno i moguće ga je prevazići koristeći dodavanje karaktera belina (blank) na kraj svih stringova čija je dužina manja od dužine stringa čija je dužina maksimalna.

```
>> a = ['M' 'a' 't' 'l' 'a' 'b']    >> b = ' Matlab'  
>> whos a                          >> whos b  
>> a = ['Matlab'; ' matlab']      >> a(1,:),    >> a(:, 1)
```

Nizovi karaktera

```
>> a = ['ponedeljak','utorak','sreda','cetvrtak';  
        'petak','subota','nedelja']  
>> a = char('ponedeljak','utorak','sreda','cetvrtak',  
            'petak','subota','nedelja')
```

Nizovi karaktera

```
>> a = ['ponedeljak'; 'utorak'; 'sreda'; 'cetvrtak';  
        'petak'; 'subota'; 'nedelja']  
>> a = char('ponedeljak', 'utorak', 'sreda', 'cetvrtak',  
            'petak', 'subota', 'nedelja')
```

Korišćenje naredbe *char* je daleko ugodniji način. Sličnu funkciju ima i funkcija *strvcat*.

Nizovi karaktera

```
>> a = ['ponedeljak'; 'utorak'; 'sreda'; 'cetvrtak';  
        'petak'; 'subota'; 'nedelja']  
>> a = char('ponedeljak', 'utorak', 'sreda', 'cetvrtak',  
            'petak', 'subota', 'nedelja')
```

Korišćenje naredbe *char* je daleko ugodniji način. Sličnu funkciju ima i funkcija *strvcat*.

Naredba koja omogućava odbacivanje karaktera beline sa kraja stringa je *deblank*.

Nizovi karaktera

```
>> a = ['ponedeljak',' utorak    ','sreda      ','cetvrtak  ';  
        'petak      ','subota    ','nedelja  ']  
>> a = char('ponedeljak',' utorak','sreda','cetvrtak',  
            'petak','subota','nedelja')
```

Korišćenje naredbe *char* je daleko ugodniji način. Sličnu funkciju ima i funkcija *strvcat*.

Naredba koja omogućava odbacivanje karaktera beline sa kraja stringa je *deblank*.

Funkcija *strtrim* izbacuje iz stringa beline na početku i kraju stringa.

Nizovi karaktera

```
>> a = ['ponedeljak',' utorak    ','sreda      ','cetvrtak  ';  
        'petak      ','subota    ','nedelja  ']  
>> a = char('ponedeljak',' utorak','sreda','cetvrtak',  
            'petak','subota','nedelja')
```

Korišćenje naredbe *char* je daleko ugodniji način. Sličnu funkciju ima i funkcija *strvcats*.

Naredba koja omogućava odbacivanje karaktera beline sa kraja stringa je *deblank*.

Funkcija *strtrim* izbacuje iz stringa beline na početku i kraju stringa.

Nizovi karaktera

Operacija konkatencije stringova, nadovezivanje stringova jedan na drugi, se može vršiti korišćenjem operatora `[]` ili korišćenjem naredbe *strcat*.

Nizovi karaktera

Operacija konkatencije stringova, nadovezivanje stringova jedan na drugi, se može vršiti korišćenjem operatora `[]` ili korišćenjem naredbe *strcat*.

```
>> a = ['MATLAB' 'matlab']  
>> b = strcat('MATLAB',' matlab')
```

Nizovi karaktera

Operacija konkatencije stringova, nadovezivanje stringova jedan na drugi, se može vršiti korišćenjem operatora `[]` ili korišćenjem naredbe *strcat*.

```
>> a = ['MATLAB' 'matlab']  
>> b = strcat('MATLAB',' matlab')
```

Funkcija *blanks* kreira string ispunjen belinama. Na primer, *blanks(5)* kreira 5 belina, dok naredba *disp(blanks(10))* kreira 10 praznih redova.

Nizovi karaktera

Operacija konkatencije stringova, nadovezivanje stringova jedan na drugi, se može vršiti korišćenjem operatora `[]` ili korišćenjem naredbe *strcat*.

```
>> a = ['MATLAB' 'matlab']  
>> b = strcat('MATLAB',' matlab')
```

Funkcija *blanks* kreira string ispunjen belinama. Na primer, *blanks(5)* kreira 5 belina, dok naredba *disp(blanks(10)')* kreira 10 praznih redova. Funkcija *lower* menja svako pojavljivanje velikog slova odgovarajućim malim slovom.

Nizovi karaktera

Operacija konkatencije stringova, nadovezivanje stringova jedan na drugi, se može vršiti korišćenjem operatora `[]` ili korišćenjem naredbe *strcat*.

```
>> a = ['MATLAB' 'matlab']  
>> b = strcat('MATLAB',' matlab')
```

Funkcija *blanks* kreira string ispunjen belinama. Na primer, *blanks(5)* kreira 5 belina, dok naredba *disp(blanks(10))* kreira 10 praznih redova. Funkcija *lower* menja svako pojavljivanje velikog slova odgovarajućim malim slovom. Funkcija *upper* menja svako pojavljivanje malog slova odgovarajućim velikim slovom.

Komparacija stringova

Komparacija stringova može se vršiti korišćenjem relacionih operatora. U ovom slučaju stringovi se tretiraju kao vektori.

Komparacija stringova

Komparacija stringova može se vršiti korišćenjem relacionih operatora. U ovom slučaju stringovi se tretiraju kao vektori.

```
>>' matlab' >=' baltam'    >> b =' Matlab'
```

Komparacija stringova

Komparacija stringova može se vršiti korišćenjem relacionih operatora. U ovom slučaju stringovi se tretiraju kao vektori.

```
>>' matlab' >=' baltam'    >> b =' Matlab'
```

<i>strcmp</i>	vrši komparaciju stringova, vraća vrednost tačno ako su stringovi identični
<i>strncmp</i>	vrši komparaciju prvih <i>n</i> karaktera, vraća vrednost tačno ako je prvih <i>n</i> karaktera u stringovima identično
<i>strcmpi</i>	deluje isto kao <i>strcmp</i> , razlika je što mala i velika slova tretira kao iste karaktere
<i>strncmpi</i>	deluje isto kao <i>strncmp</i> , razlika je što mala i velika slova tretira kao iste karaktere

Pretraga stringa i zamena podstringova

<i>strrep</i>	vrši zamenu podstringa drugim stringom u zadatom stringu
<i>findstr</i>	pronalaži poziciju stringa u zadatom stringu
<i>strtok</i>	vraća podstring koji se nalazi na početku stringa a koji se završava prvom pojavom završnog karaktera obično grupa belina
<i>strmatch</i>	vraća podstring koji počinje zadatim stringom

Pretraga stringa i zamena podstringova

<i>strrep</i>	vrši zamenu podstringa drugim stringom u zadatom stringu
<i>findstr</i>	pronalaži poziciju stringa u zadatom stringu
<i>strtok</i>	vraća podstring koji se nalazi na početku stringa a koji se završava prvom pojavom završnog karaktera obično grupa belina
<i>strmatch</i>	vraća podstring koji počinje zadatim stringom

```
>> datum = '10/10/1972';  
>> findstr('10', datum)           >> findstr(datum, '10')  
>> strrep(datum, '10', '09')  
>> datumi = [datum ' ' datum];    >> strtok(datumi)  
>> strtok(datumi, '\')
```

Merenje vremenskih intervala

Funkcije *cputime*, *tic* i *toc* mogu se koristiti za merenje vremena.

Merenje vremenskih intervala

Funkcije *cputime*, *tic* i *toc* mogu se koristiti za merenje vremena. Funkcije *tic* i *toc* se koriste za merenje proteklog vremena između dva trenutka u izvršenju. Funkcija *tic* postavlja početni trenutak, dok funkcija *toc* postavlja krajnji trenutak merenja vremena.

Merenje vremenskih intervala

Funkcije *cputime*, *tic* i *toc* mogu se koristiti za merenje vremena. Funkcije *tic* i *toc* se koriste za merenje proteklog vremena između dva trenutka u izvršenju. Funkcija *tic* postavlja početni trenutak, dok funkcija *toc* postavlja krajnji trenutak merenja vremena.

```
>> tic; A = cell(10000, 10000); B = cell(10000, 10000); toc
```

Nizovi ćelija

Tip podataka matrica nije u stanju da ima elemente koji su različitog tipa. Ovakva funkcionalnost se postiže nizovima ćelija.

Nizovi ćelija

Tip podataka matrica nije u stanju da ima elemente koji su različitog tipa. Ovakva funkcionalnost se postiže nizovima ćelija. Da bismo kreirali niz ćelija A tipa 2×2 , možemo postupiti na sledeći način.

Nizovi ćelija

Tip podataka matrica nije u stanju da ima elemente koji su različitog tipa. Ovakva funkcionalnost se postiže nizovima ćelija. Da bismo kreirali niz ćelija A tipa 2×2 , možemo postupiti na sledeći način.

```
>> A(1, 1) = {[1 2 3; 4 5 6; 7 8 9]}    >> A(1, 2) = {'Joe Tribiani'}  
>> A(2, 2) = {uint16(1)}                >> A(2, 1) = {2 * i}
```

Prilikom kreiranja niza ćelija, morali smo da elemente niza ćelija stavimo u zagrade {}, da bismo istakli činjenicu da kreiramo niz ćelija.

Nizovi ćelija

Tip podataka matrica nije u stanju da ima elemente koji su različitog tipa. Ovakva funkcionalnost se postiže nizovima ćelija. Da bismo kreirali niz ćelija A tipa 2×2 , možemo postupiti na sledeći način.

```
>> A(1,1) = {[1 2 3; 4 5 6; 7 8 9]}    >> A(1,2) = {'Joe Tribiani'}  
>> A(2,2) = {uint16(1)}                >> A(2,1) = {2 * i}
```

Prilikom kreiranja niza ćelija, morali smo da elemente niza ćelija stavimo u zagrade {}, da bismo istakli činjenicu da kreiramo niz ćelija. Da pristupimo pojedinim elementima niza ćelija koristimo na primer `>> A(1,2)`.

Nizovi ćelija

Drugi način za kreiranje nizova ćelija koristi zagrade $\{\}$ za indekse, a ne za elemente.

Nizovi ćelija

Drugi način za kreiranje nizova ćelija koristi zagrade $\{\}$ za indekse, a ne za elemente.

```
>> B{1,1} = [1 2 3; 4 5 6; 7 8 9]    >> A{1,2} = 'Joe Tribiani'  
>> A{2,1} = uint16{1}                >> A{2,2} = 2 * i
```

Nizovi ćelija

Drugi način za kreiranje nizova ćelija koristi zagrade $\{\}$ za indekse, a ne za elemente.

```
>> B{1,1} = [1 2 3; 4 5 6; 7 8 9]    >> A{1,2} = 'Joe Tribiani'  
>> A{2,1} = uint16{1}                >> A{2,2} = 2 * i
```

Prazan niz ćelija ima posebnu notaciju $\{\}$.