

Wektory i adresowanie [6/8]

- Wektory reprezentujące regularne sekwencje liczb można wygodnie tworzyć za pomocą komendy:

:

- Na przykład tak:

```
>> na_przykład_tak = 1:6
```

- Albo, jeżeli chcemy sekwencję rosnącą o 2 z każdym elementem:

```
>> na_przykład_inaczej = 1:2:9
```

- Ogólna zasada jest taka:

```
>> jakas_sekwencja = start:co_ile:koniec
```

Wektory i adresowanie [7/8]

■ Adresować można jeszcze w sposób logiczny:

```
>> inny_wektor = [4, 7, 1, 3, 8, 2];
```

```
>> inny_wektor[inny_wektor > 2]
```

■ Może wygląda to trochę dziwnie ale zobaczcie co jest wynikiem komendy

```
>> inny_wektor > 2
```

```
ans =
```

```
1    1    0    1    1    0
```

■ Dostajemy w rezultacie wektor zer i jedynek - faktycznie są to wartości fałsz (0) oraz prawda (1)

Wektory i adresowanie [8/8]

- Jeżeli adresujemy wektorem logicznym dostajemy elementy odpowiadające wartościom prawdziwym wektora adresującego:

1	1	0	1	1	0
[4,	7,	1,	3,	8,	2]

Macierze i ich adresowanie

- Z macierzami jest podobnie jak z wektorami tyle, że wektory mają tylko jeden wymiar, natomiast macierze – dwa lub więcej:

wektor

3	7	2	9	4
1	2	3	4	5

macierz

	1	2	3	4	5
1	3	7	2	9	4
2	1	2	2	8	4
3	3	5	4	9	11
4	2	6	2	9	4

Macierze i ich adresowanie

► Macierz możemy utworzyć podobnie jak wektor:

```
>> moja_macierz1 = [1, 2, 3; 4, 5, 6; 7, 8, 9]
```

```
>> moja_macierz2 = [1, 4, 7; 2, 5, 8; 3, 6, 9]
```

```
>> moja_macierz2 = [1, 4, 7;  
                    2, 5, 8;  
                    3, 6, 9]
```

► Ale możemy macierze tworzyć też tak:

```
>> losowa_macierz = rand(5,5)
```

```
>> same_zera = zeros(4,6)
```

Macierze i ich adresowanie

- Macierze adresujemy podobnie jak wektory, tyle że musimy podać adres dla każdego wymiaru macierzy:

```
>> moja_macierz = [1, 2, 3; 4, 5, 6; 7, 8, 9]
```

```
>> moja_macierz(2,3)
```

- 0 adresowaniu macierzy dwu-wymiarowych warto myśleć jak o szukaniu swojego miejsca w kinie.
- Wchodząc do sali kinowej najpierw szukamy **rzędu**, w którym mamy miejsca, dopiero potem szukamy **miejsca**.

ekran

	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

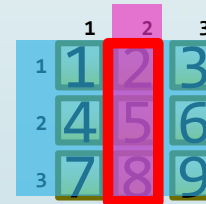
Macierze i ich adresowanie

- Wszystkie zasady dotyczące adresowania wektorów odnoszą się też do adresowania macierzy:

```
>> moja_macierz(3, [1 3])
```

- Gdy chcemy uzyskać wszystkie elementy z drugiej kolumny możemy napisać:

```
>> moja_macierz(:, 2)
```

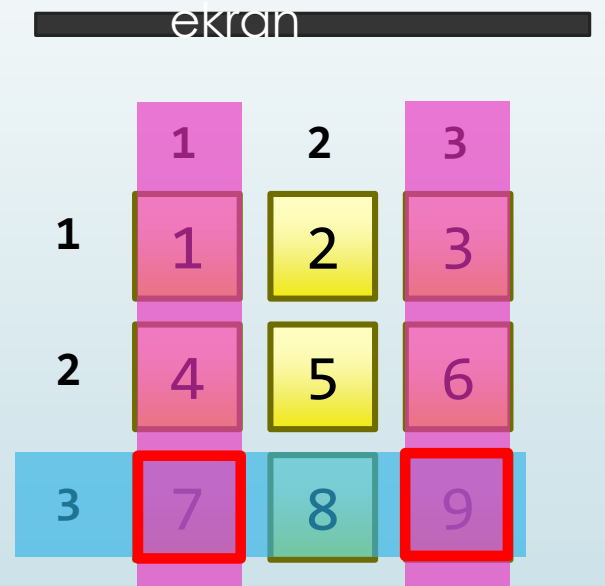


A 3x3 matrix with elements 1 to 9. The second column (elements 2, 5, 8) is highlighted with a red border. The columns are labeled 1, 2, 3 at the top, and the rows are labeled 1, 2, 3 on the left.

	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

- I odwrotnie:

```
>> moja_macierz(2, :)
```



A 3x3 matrix with elements 1 to 9. The second row (elements 4, 5, 6) is highlighted with a red border. The columns are labeled 1, 2, 3 at the top, and the rows are labeled 1, 2, 3 on the left.

	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

Macierze i ich adresowanie

► Kolejne zadanie :

Stwórzcie za pomocą komendy `rand` macierz mającą 8 rzędów oraz 15 kolumn.

Zaadresujcie element znajdujący się na przecięciu 3 rzędu i 11 kolumny.

Następnie zaadresujcie wszystkie elementy znajdujące się na przecięciu kolumn: 2, 5, 13 oraz wierszy 5 i 7.

Ostatnie zadanie: weź wszystkie elementy na przecięciu parzystych wierszy i nieparzystych kolumn.

