

Piotr Chojnacki:

MATLAB

zajęcia 1:

{Poniżej znajduje się zrzut ekranowy pola gdzie wpisuje się komendy. Nie wszystkie komendy są opisane. Wynik działania każdej z nich widać linijkę niżej. Znajdziesz tutaj podstawowe komendy i operacje na macierzach. Komendy należy wpisywać w Workspace i potwierdzając enterem.}

zajecia1.mat

>> **A=[1,2,3;4,5,6;7,8,9]** {komenda powodująca wypisanie macierzy o trzech wierszach i trzech kolumnach}

A = {macierz A}

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

>> **x=[2:10]** {wprowadzenie danych poprzez wygenerowanie z przedziału od 2 do 10 co 1}

x =

```
2  3  4  5  6  7  8  9  10
```

>> **y=[2:5:10]** {wprowadzenie danych poprzez wygenerowanie z przedziału od 2 do 10 co 5}

y =

```
2  7
```

>> **rand(5)** {wprowadzenie danych poprzez wygenerowanie losowo macierzy o wielkości 5 wierszy na 5 kolumn z przedziału od 0 do 1}

ans =

```
0.9501  0.7621  0.6154  0.4057  0.0579
0.2311  0.4565  0.7919  0.9355  0.3529
0.6068  0.0185  0.9218  0.9169  0.8132
0.4860  0.8214  0.7382  0.4103  0.0099
0.8913  0.4447  0.1763  0.8936  0.1389
```

>> **round(rand(5))** {komenda ta powoduje wprowadzenie danych poprzez wygenerowanie macierzy z przedziału <0,1> a następnie zaokrąglenie do samych 1 i 0. Dla każdej liczby losowej większej od 0,5 wychodzi 1 a dla mniejszej 0.}

ans =

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

>> `round(rand(5)*20)` {komenda ta powoduje wprowadzenie danych poprzez wygenerowanie macierzy 5 x 5 dla liczb z przedziału od 0 do 20}

ans =

```
4 14 10 13 15
14 8 18 7 6
6 17 16 6 17
11 17 13 7 11
3 12 16 11 7
```

>> `round(rand(5,10)*20)` {komenda ta powoduje wprowadzenie danych poprzez wygenerowanie macierzy 5 wierszy x 10 kolumn dla liczb z przedziału od 0 do 20}

ans =

```
14 16 20 3 13 12 5 4 11 8
11 19 5 0 6 8 12 8 16 6
9 10 5 18 9 10 15 16 1 17
14 18 18 4 1 7 11 14 12 0
12 3 15 6 20 9 13 9 1 15
```

>> `round(rand(5,10))*20` {komenda ta powoduje wprowadzenie danych poprzez wygenerowanie macierzy 5 wierszy x 10 kolumn. Ale najpierw jest generowana (z przedziału <0.1>) i zaokrąglana macierz do 0 lub 1. Następnie wszystko jest mnożone przez 20 i tam gdzie była 1 jest 20, a tam gdzie 0 jest 0.}

ans =

```
20 0 0 20 0 0 0 20 0 0
20 20 20 0 20 0 20 0 0 0
20 0 0 20 20 0 0 0 20 0
0 20 0 20 0 0 0 0 20 0
0 20 20 0 20 0 20 20 20 0
```

>> `round((rand(5,10)-0.5)*(2*10))` {macierz z przedziału od -10 do 10}

ans =

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

```
5 4 -5 -9 -7 -4 -9 -7 3 -1
0 8 6 -8 -7 -3 -1 7 7 8
1 -5 8 3 10 -2 7 -5 -10 5
```

```
>> round((rand(5,10))*20)-10
```

```
ans =
```

```
4 -2 -1 -4 1 -2 8 -6 1 3
-3 7 -1 -9 -2 -10 7 8 9 4
-7 0 -2 4 -6 -2 -3 1 -3 -2
-7 6 8 3 3 5 2 3 3 -2
-6 -1 -10 10 5 6 5 -5 -2 3
```

```
>> round((rand(5,10))*30)-10
```

```
ans =
```

```
15 11 11 -7 15 1 7 3 17 12
1 5 19 10 10 6 11 9 8 3
3 13 14 1 20 -2 19 14 -2 19
8 5 11 -6 19 8 13 -7 16 -8
7 -4 5 7 -8 -9 12 18 5 7
```

```
>> ones(20) {macierz o rozmiarze n x n, w tym przypadku 20 x 20 o wszystkich elementach
równych 1}
```

```
ans =
```

```
Columns 1 through 14
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Columns 15 through 20

```
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
```

>> **ones(2)** {macierz o rozmiarze **n x n**, w tym przypadku 2 x 2 o wszystkich elementach równych **1**}

ans =

```
1 1
1 1
```

>> **zeros(4)** {wygenerowanie macierzy samych zer o liczbie wierszy 4 i liczbie kolumn 4}

ans =

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

>> **pascal(5)** {wygenerowanie macierzy o liczbie wierszy 4 i liczbie kolumn 4, w której liczby sumują się po przekatnych. Bardzo łatwo zauważyć tę zależność.}

ans =

```
1 1 1 1 1
1 2 3 4 5
```

```
1 3 6 10 15
1 4 10 20 35
1 5 15 35 70
```

```
>> magic(7) {wygenerowanie macierzy losowej o liczbie wierszy 7 i liczbie kolumn 7}
```

```
ans =
```

```
30 39 48 1 10 19 28
38 47 7 9 18 27 29
46 6 8 17 26 35 37
5 14 16 25 34 36 45
13 15 24 33 42 44 4
21 23 32 41 43 3 12
22 31 40 49 2 11 20
```

```
>> magic(2) {wygenerowanie macierzy losowej o liczbie wierszy 2 i liczbie kolumn 2}
```

```
ans =
```

```
1 3
4 2
```

```
>> eye(2)
```

```
ans =
```

```
1 0
0 1
```

```
>> eye(2,3)
```

```
ans =
```

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

```
>> eye(2,5)
```

```
ans =
```

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

```
>> pascal(5,2)
```

```
ans =
```

```
1 1 1 1 1
-4 -3 -2 -1 0
6 3 1 0 0
```

```
-4 -1 0 0 0
1 0 0 0 0
```

>> **magic(4)** {wygenerowanie macierzy kwadratowej o liczbie wierszy n i liczbie kolumn n. Dlatego też poprzednie komendy nie działały bo za każdym razem próbowałem wypisać macierz niekwadratową.}

ans =

```
16 2 3 13
5 11 10 8
9 7 6 12
4 14 15 1
```

>> **save** {zapisywanie danych do zbioru o nazwie matlab.mat, nazwa autoamtyczna}

Saving to: matlab.mat

>> **clear** {czyszczenie danych}

>> **load** {wczytywanie danych do programu}

Loading from: matlab.mat

>> **save zajecia** {zapisywanie danych do zbioru o nazwie zajecia.mat}

>> **save zajecia1**

>> **save zajecia1**

>> **save wynik.txt** {zapisywanie danych do zbioru o nazwie wynik.txt w ascii}

>> **save zajecia**

>> **save(matlab.mat)**

{poniższe wyniki obliczeń tzn. format wyświetlania wyniku, liczbe miejsc po przecinku można zmieniać w preferencjach programu. File -> preferences... -> comandnd window-> (zmieniać numeric format)}

>> **1/3**

ans =

```
0.3333
```

>> **1/3**

ans =

```
0.3333333333333333
```

>> **1/3**

ans =

3.3333e-001

>> 1/3

ans =

0.33333

>> 1/3

ans =

0.3333

>> save zajecia1

>> 1/3

ans =

1/3

>> 1/3

ans =

+

>> -0,2

ans =

ans =

+

>> -2

ans =

-

>> 3

ans =

+

```
>>  
>> 3
```

```
ans =
```

```
+
```

```
>> exp(1) {wyświetlenie wartości liczby e, natomiast zmieniając w Command Window  
numeric format można uzyskać dokładniejsze wartości}
```

```
ans =
```

```
2.7183
```

```
>> exp(1)
```

```
ans =
```

```
2.71828182845905
```

```
>> pi
```

```
ans =
```

```
3.14159265358979
```

```
>> magic(5) {wygenerowanie macierzy losowej o liczbie wierszy 5 i liczbie kolumn 5}
```

```
ans =
```

```
17 24 1 8 15  
23 5 7 14 16  
4 6 13 20 22  
10 12 19 21 3  
11 18 25 2 9
```

```
>> A=A+1 {podstawienie zmiennej}
```

```
A =
```

```
2 3 4  
5 6 7  
8 9 10
```

```
>> A=A+123
```

```
A =
```

```
125 126 127  
128 129 130  
131 132 133
```



```
>> A=A+123/52
```

```
A =
```

```
1.0e+002 *
```

```
1.27365384615385 1.28365384615385 1.29365384615385  
1.30365384615385 1.31365384615385 1.32365384615385  
1.33365384615385 1.34365384615385 1.35365384615385
```

```
>> B=[24,52;58,54;69,69]
```

```
B =
```

```
24 52  
58 54  
69 69
```

```
>> B=[24,52;58,54;69,69;58,69]
```

```
B =
```

```
24 52  
58 54  
69 69  
58 69
```

```
>> B=[24,52,52;58,54,58;69,69,47]
```

```
B =
```

```
24 52 52  
58 54 58  
69 69 47
```

```
>> A=A+B
```

```
A =
```

```
1.0e+002 *
```

```
1.51365384615385 1.80365384615385 1.81365384615385  
1.88365384615385 1.85365384615385 1.90365384615385  
2.02365384615385 2.03365384615385 1.82365384615385
```

```
>> A=A+B
```

```
A =
```

```
175.3654 232.3654 233.3654
```

246.3654 239.3654 248.3654
271.3654 272.3654 229.3654

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

A =

1 2 3
4 5 6
7 8 9

>> B=[24,52;58,54;69,69;58,69]

B =

24 52
58 54
69 69
58 69

>> B=[24,52;58,54;69,69;58,69]

B =

24 52
58 54
69 69
58 69

>> B=[24,52,52;58,54,58;69,69,47]

B =

24 52 52
58 54 58
69 69 47

>> A=A+B

A =

25 54 55
62 59 64
76 77 56

>> A=A+B

A =

49 106 107
120 113 122

```
145 146 103
>> A
A =
    49 106 107
    120 113 122
    145 146 103
>> A
A =
    49 106 107
    120 113 122
    145 146 103
>> A(2,2)
ans =
    113
>> A(2,:)
ans =
    120 113 122
>> A(2,2:3)
ans =
    113 122
>> A(2,2:3)
ans =
    113 122
>>
>> A
A =
    49 106 107
    120 113 122
```

```

145 146 103

>> A

A =

    49 106 107
    120 113 122
    145 146 103

>> A(1,:)

ans =

    49 106 107

>> A

A =

    49 106 107
    120 113 122
    145 146 103

>> A(1,:)=[]

A =

    120 113 122
    145 146 103

>> A(:,2)=[]

A =

    120 122
    145 103

>> 1:5

ans =

     1     2     3     4     5

>> save zajecial
>> 1:2:10

ans =

     1     3     5     7     9

```

```
>> 2:2:10
```

```
ans =
```

```
    2    4    6    8   10  
>> 2:3:10
```

```
ans =
```

```
    2    5    8
```

```
>> 2:4:10
```

```
ans =
```

```
    2    6   10
```

```
>> W=[1:5]
```

```
W =
```

```
    1    2    3    4    5
```

```
>> C=[1:9]
```

```
C =
```

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

```
>> D=[W C]
```

```
D =
```

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

```
>> save zajecia1
```

```
>> A
```

```
A =
```

```
   120   122  
   145   103
```

```
>> B
```

```
B =
```

```
   24   52   52  
   58   54   58
```

69 69 47

```
>> B(:,3)=[] {usunięcie 3 kolumny z macierzy B}
```

B =

```
24 52
58 54
69 69
```

```
>> B(3,:)=[]
```

B =

```
24 52
58 54
```

```
>> A(:,2)=B(2,2)
```

A =

```
120 54
145 54
```

```
>> A
```

A =

```
120 54
145 54
```

```
>> B
```

B =

```
24 52
58 54
```

```
>> B(1,:)= [7,7]
```

B =

```
7 7
58 54
```

```
>> A
```

A =

```
120 54
145 54
```

```
>> B
```

```
B =
```

```
7 7
58 54
```

```
>>
```

zajecia2.mat

```
>> A=magic(6) {wygenerowanie macierzy A kwadratowej o liczbie wierszy 6 i liczbie kolumn 6.}
```

```
A =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> B=magic(4) {wygenerowanie macierzy B kwadratowej o liczbie wierszy 4 i liczbie kolumn 4.}
```

```
B =
```

```
16 2 3 13
5 11 10 8
9 7 6 12
4 14 15 1
```

```
>> sort(B) {komenda sortująca macierz B kolejno według liczb w wierszach}
```

```
ans =
```

```
4 2 3 1
5 7 6 8
9 11 10 12
```

16 14 15 13

>> sort(B') {sortowanie transponowanej macierzy B}

ans =

2 5 6 1
3 8 7 4
13 10 9 14
16 11 12 15

>> sum(B) {suma macierzy B po kolumnach}

ans =

34 34 34 34

>> B {wypisanie macierzy B. Jest to częste polecenie w tym pliku, aby przypominać sobie pierwotny kształt macierzy B, więc nie będzie to polecenie więcej opisywane}

B =

16 2 3 13
5 11 10 8
9 7 6 12
4 14 15 1

>> sort(B)

ans =

4 2 3 1
5 7 6 8
9 11 10 12
16 14 15 13

>> sort(B')

ans =

2 5 6 1
3 8 7 4
13 10 9 14
16 11 12 15

>> (sort(B'))' {transponowanie poprzednio posortowanej transponowanej macierzy B. Czyli posortowanie w wierszach}

ans =

2 3 13 16


```
5 8 10 11
6 7 9 12
1 4 14 15
```

```
>> save zajecia2
```

```
>> sum(sum(B)) {suma zsumowanych kolumn macierzy B. Suma kolumn wynosiła kolejno 34 34 34 34, a więc ich suma daje 136}
```

```
ans =
```

```
136
```

```
>> prod(B) {iloczyn wartości kolumn}
```

```
ans =
```

```
2880 2156 2700 1248
```

```
>> cumsum(B) {sumowanie po kolumnach, ale w trochę inny sposób. Pierwszy wyraz zostaje bez zmian i do niego dodawany jest następny zastępując jego miejsce.}
```

```
ans =
```

```
16 2 3 13
21 13 13 21
30 20 19 33
34 34 34 34
```

```
>> B
```

```
B =
```

```
16 2 3 13
5 11 10 8
9 7 6 12
4 14 15 1
```

```
>> cumsum(B)
```

```
ans =
```

```
16 2 3 13
21 13 13 21
30 20 19 33
34 34 34 34
```

```
>> save zajecia2
```

```
>> cumprod(B) {iloczyn po kolumnach, ale w trochę inny sposób. Pierwszy wyraz zostaje bez zmian, a następnie jest on mnożony przez następny zastępując jego miejsce.}
```

ans =

```
16    2    3    13
80    22   30   104
720   154  180  1248
2880  2156 2700 1248
```

>> B

B =

```
16  2  3  13
5  11 10  8
9  7  6  12
4  14 15  1
```

>> **diag(B)** {komenda wypisująca przekątną macierzy}

ans =

```
16
11
6
1
```

>> **diag(diag(B))** {komenda pozostawiająca przekątną macierzy. Reszta liczb jest równa zero}

ans =

```
16  0  0  0
0  11 0  0
0  0  6  0
0  0  0  1
```

>> A

A =

```
35  1  6  26  19  24
3  32  7  21  23  25
31  9  2  22  27  20
8  28 33  17  10  15
30  5 34  12  14  16
4  36 29  13  18  11
```

>> **flipud(A)** {komenda zamieniająca kolejność wierszy}

ans =

```
4 36 29 13 18 11
30 5 34 12 14 16
8 28 33 17 10 15
31 9 2 22 27 20
3 32 7 21 23 25
35 1 6 26 19 24
```

```
>> flipr(A) {komenda zamieniająca kolejność kolumn}
```

```
ans =
```

```
24 19 26 6 1 35
25 23 21 7 32 3
20 27 22 2 9 31
15 10 17 33 28 8
16 14 12 34 5 30
11 18 13 29 36 4
```

```
>> A(-1:1,:)

```

```
ans =
```

```
Empty matrix: 0-by-6
```

```
>> A(1:-1:1,:)

```

```
ans =
```

```
35 1 6 26 19 24
```

```
>> A
```

```
A =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> A
```

```
A =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
```

```
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> A(3:-1:2,:) {komanda zamieniająca miejscami 3 i 2 wiersz}
```

```
ans =
```

```
31 9 2 22 27 20
3 32 7 21 23 25
```

```
>> A
```

```
A =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> tril(A) {wybór górnej połówki macierzy A}
```

```
ans =
```

```
35 0 0 0 0 0
3 32 0 0 0 0
31 9 2 0 0 0
8 28 33 17 0 0
30 5 34 12 14 0
4 36 29 13 18 11
```

```
>> A
```

```
A =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> triu(A) {wybór dolnej połówki macierzy A}
```

```
ans =
```

```
35  1  6 26 19 24
 0 32  7 21 23 25
 0  0  2 22 27 20
 0  0  0 17 10 15
 0  0  0  0 14 16
 0  0  0  0  0 11
```

```
>> tril(A,4)
```

```
ans =
```

```
35  1  6 26 19  0
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> A
```

```
A =
```

```
35  1  6 26 19 24
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> tril(A,2)
```

```
ans =
```

```
35  1  6  0  0  0
 3 32  7 21  0  0
31  9  2 22 27  0
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> tril(A,1)
```

```
ans =
```

```
35  1  0  0  0  0
 3 32  7  0  0  0
31  9  2 22  0  0
 8 28 33 17 10  0
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> A
```

```
A =
```

```
35  1  6 26 19 24
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> A=magic(6)
```

```
A =
```

```
35  1  6 26 19 24
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> min(A) {minimum macierzy A według kolumn}
```

```
ans =
```

```
3  1  2 12 10 11
```

```
>> max(A) {maximum macierzy A według kolumn}
```

```
ans =
```

```
35 36 34 26 27 25
```

```
>> [pmin xmax]=min(A) {zestaw poleceń pokazujących położenie najmniejszego lub największego elementu}
```

```
pmin =
```

```
3  1  2 12 10 11
```

```
xmax =
```

```
2  1  3  5  4  6
```

```
>> A
```

```
A =
```

```
35  1  6 26 19 24
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> min(min(A))
```

```
ans =
```

```
1
```

```
>> max(max(A))
```

```
ans =
```

```
36
```

```
>> [pmin xmax]=min(min(A))
```

```
pmin =
```

```
1
```

```
xmax =
```

```
2
```

```
>> A
```

```
A =
```

```
35  1  6 26 19 24
 3 32  7 21 23 25
31  9  2 22 27 20
 8 28 33 17 10 15
30  5 34 12 14 16
 4 36 29 13 18 11
```

```
>> [pmin xmax]=min(min(A'))
```

```
pmin =
```

```
1
```

```
xmax =
```

```
1
```

```
>> mean(A) {średnia arytmetyczna danych elementów}
```

```
ans =
```

```
18.5000 18.5000 18.5000 18.5000 18.5000 18.5000
```

```
>> mean(B)
```

```
ans =
```

```
8.5000 8.5000 8.5000 8.5000
```

```
>> B
```

```
B =
```

```
16  2  3 13  
 5 11 10  8  
 9  7  6 12  
 4 14 15  1
```

```
>> save zajecia2
```

```
>> C=B(:) {podstawienie za C kolejnych kolumn. Jedna za druga}
```

```
C =
```

```
16  
 5  
 9  
 4  
 2  
11  
 7  
14  
 3  
10  
 6  
15  
13  
 8  
12  
 1
```

```
>> B
```

```
B =
```

```
16  2  3 13  
 5 11 10  8  
 9  7  6 12
```



```

4 14 15 1
>> save zajecia2
>> D=[1,2,4,8,11,2,0,1]
D =
    1    2    4    8   11    2    0    1
>> E=[6,12,47,2,0,1,58,1,0]
E =
    6   12   47    2    0    1   58    1    0
>> E=[6,12,47,2,0,1,1,0]
E =
    6   12   47    2    0    1    1    0
>> D
D =
    1    2    4    8   11    2    0    1
>> E
E =
    6   12   47    2    0    1    1    0
>> save zajecia2
>> D=D-4 {odjęcie 4 od wszystkich elementów macierzy}
D =
   -3  -2    0    4    7  -2  -4  -3
>> E=E-4
E =
    2    8   43   -2   -4   -3   -3   -4
>> D
D =

```

```
-3 -2 0 4 7 -2 -4 -3
```

```
>> E
```

```
E =
```

```
2 8 43 -2 -4 -3 -3 -4
```

```
>> D==E {porównanie czy elementy są sobie równe. Tam gdzie warunek spełniony wypisana zostaje 1, a gdzie nie to 0}
```

```
ans =
```

```
0 0 0 0 0 0 0 0
```

```
>> D==D
```

```
ans =
```

```
1 1 1 1 1 1 1 1
```

```
>> D<E {czy elementy są mniejsze Tam gdzie warunek spełniony wypisana zostaje 1, a gdzie nie to 0}
```

```
ans =
```

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

```
>> D<=E {Czy elementy są mniejsze równe. Tam gdzie warunek spełniony wypisana zostaje 1, a gdzie nie to 0}
```

```
ans =
```

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

```
>> D>=E {Czy elementy są większe równe. Tam gdzie warunek spełniony wypisana zostaje 1, a gdzie nie to 0}
```

```
ans =
```

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

```
>> D~E {Czy elementy są mniejsze różne. Tam gdzie warunek spełniony wypisana zostaje 1, a gdzie nie to 0}
```

```
ans =
```

```
1 1 1 1 1 1 1 1
```

```
>> save zajecia2
```

```
>> D|E {alternatywa}
```

```
ans =
```

```
1 1 1 1 1 1 1 1
```

```
>> D&E {koniunkcja}
```

```
ans =
```

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

```
>> ~E {negacja}
```

```
ans =
```

```
0 0 0 0 0 0 0 0
```

```
>> any(A)
```

```
ans =
```

```
1 1 1 1 1 1
```

```
>> all(E)
```

```
ans =
```

```
1
```

```
>> D
```

```
D =
```

```
-3 -2 0 4 7 -2 -4 -3
```

```
>> W1=D>2 {wyświetlenie jedynie pozycji elementów spełniających warunek}
```

```
W1 =
```

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

```
>> W2=D(D>2) {wypisanie elementów spełniających warunek}
```

```
W2 =
```

```
4 7
```

```
>> W3=D.*(D>2) {wypisanie elementów spełniających warunek na odpowiednich pozycjach}
```

```
W3 =
```

```
0 0 0 4 7 0 0 0
```

```
>> D
```

```
D =
```

```
-3 -2 0 4 7 -2 -4 -3
```

```
>> W4=(D>2 & D<-2)
```

```
W4 =
```

```
0 0 0 0 0 0 0 0
```

```
>> W4=(D<7 & D>0) {wyświetlenie jedynie pozycji elementów spełniających warunki}
```

```
W4 =
```

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

```
>> W4=D(D<7 & D>0) {wyświetlenie jedynie elementów spełniających warunki}
```

```
W4 =
```

```
4
```

```
>> W4=D.*(D<7 & D>0) {wyświetlenie elementów spełniających warunki na odpowiednich pozycjach}
```

```
W4 =
```

```
0 0 0 4 0 0 0 0
```

```
>> save zajecia2
```

```
>> A
```

```
A =
```

```
35 1 6 26 19 24  
3 32 7 21 23 25  
31 9 2 22 27 20  
8 28 33 17 10 15  
30 5 34 12 14 16  
4 36 29 13 18 11
```

```
>> A=A-20
```

A =

```
15 -19 -14 6 -1 4
-17 12 -13 1 3 5
11 -11 -18 2 7 0
-12 8 13 -3 -10 -5
10 -15 14 -8 -6 -4
-16 16 9 -7 -2 -9
```

>> A=A+20

A =

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

>> A=A-10

A =

```
25 -9 -4 16 9 14
-7 22 -3 11 13 15
21 -1 -8 12 17 10
-2 18 23 7 0 5
20 -5 24 2 4 6
-6 26 19 3 8 1
```

>> A=A+10

A =

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

>> W=A+10

W =

```
45 11 16 36 29 34
13 42 17 31 33 35
41 19 12 32 37 30
18 38 43 27 20 25
```

```
40 15 44 22 24 26
14 46 39 23 28 21
```

```
>> W=A-20
```

```
W =
```

```
15 -19 -14 6 -1 4
-17 12 -13 1 3 5
11 -11 -18 2 7 0
-12 8 13 -3 -10 -5
10 -15 14 -8 -6 -4
-16 16 9 -7 -2 -9
```

```
>> W=A+15
```

```
W =
```

```
50 16 21 41 34 39
18 47 22 36 38 40
46 24 17 37 42 35
23 43 48 32 25 30
45 20 49 27 29 31
19 51 44 28 33 26
```

```
>> W=A-15
```

```
W =
```

```
20 -14 -9 11 4 9
-12 17 -8 6 8 10
16 -6 -13 7 12 5
-7 13 18 2 -5 0
15 -10 19 -3 -1 1
-11 21 14 -2 3 -4
```

```
>> W=A+15
```

```
W =
```

```
50 16 21 41 34 39
18 47 22 36 38 40
46 24 17 37 42 35
23 43 48 32 25 30
45 20 49 27 29 31
19 51 44 28 33 26
```

```
>> W=A-8
```

```
W =
```

```
27 -7 -2 18 11 16
-5 24 -1 13 15 17
23 1 -6 14 19 12
0 20 25 9 2 7
22 -3 26 4 6 8
-4 28 21 5 10 3
```

```
>> W=A-2
```

```
W =
```

```
33 -1 4 24 17 22
1 30 5 19 21 23
29 7 0 20 25 18
6 26 31 15 8 13
28 3 32 10 12 14
2 34 27 11 16 9
```

```
>> W=W-2
```

```
W =
```

```
31 -3 2 22 15 20
-1 28 3 17 19 21
27 5 -2 18 23 16
4 24 29 13 6 11
26 1 30 8 10 12
0 32 25 9 14 7
```

```
>> W=W-2
```

```
W =
```

```
29 -5 0 20 13 18
-3 26 1 15 17 19
25 3 -4 16 21 14
2 22 27 11 4 9
24 -1 28 6 8 10
-2 30 23 7 12 5
```

```
>> W=W-2
```

```
W =
```

```
27 -7 -2 18 11 16
-5 24 -1 13 15 17
23 1 -6 14 19 12
0 20 25 9 2 7
22 -3 26 4 6 8
-4 28 21 5 10 3
```

```
>> W=W-2
```

```
W =
```

```
25 -9 -4 16 9 14
-7 22 -3 11 13 15
21 -1 -8 12 17 10
-2 18 23 7 0 5
20 -5 24 2 4 6
-6 26 19 3 8 1
```

```
>> W=W-2
```

```
W =
```

```
23 -11 -6 14 7 12
-9 20 -5 9 11 13
19 -3 -10 10 15 8
-4 16 21 5 -2 3
18 -7 22 0 2 4
-8 24 17 1 6 -1
```

```
>> W=W-2
```

```
W =
```

```
21 -13 -8 12 5 10
-11 18 -7 7 9 11
17 -5 -12 8 13 6
-6 14 19 3 -4 1
16 -9 20 -2 0 2
-10 22 15 -1 4 -3
```

```
>> M=all(W>0)
```

```
M =
```

```
0 0 0 0 0 0
```

```
>> M=all(W>1)
```

```
M =
```

```
0 0 0 0 0 0
```

```
>> M=W(W>1)
```

```
M =
```


21
17
16
18
14
22
19
20
15
12
7
8
3
5
9
13
4
10
11
6
2

>> M=W(W<0)

M =

-11
-6
-10
-13
-5
-9
-8
-7
-12
-2
-1
-4
-3

>> W

W =

21	-13	-8	12	5	10
-11	18	-7	7	9	11
17	-5	-12	8	13	6
-6	14	19	3	-4	1
16	-9	20	-2	0	2
-10	22	15	-1	4	-3

```
>> N=M(:)
```

```
N =
```

```
-11  
-6  
-10  
-13  
-5  
-9  
-8  
-7  
-12  
-2  
-1  
-4  
-3
```

```
>> M
```

```
M =
```

```
-11  
-6  
-10  
-13  
-5  
-9  
-8  
-7  
-12  
-2  
-1  
-4  
-3
```

```
>> N=M(:)
```

```
N =
```

```
-11  
-6  
-10  
-13  
-5  
-9  
-8  
-7  
-12  
-2  
-1
```

```
-4  
-3
```

```
N =
```

```
-11 -6 -10 -13 -5 -9 -8 -7 -12 -2 -1 -4 -3
```

```
>> V=[M M]
```

```
V =
```

```
-11 -11  
-6 -6  
-10 -10  
-13 -13  
-5 -5  
-9 -9  
-8 -8  
-7 -7  
-12 -12  
-2 -2  
-1 -1  
-4 -4  
-3 -3
```

```
>> W
```

```
W =
```

```
21 -13 -8 12 5 10  
-11 18 -7 7 9 11  
17 -5 -12 8 13 6  
-6 14 19 3 -4 1  
16 -9 20 -2 0 2  
-10 22 15 -1 4 -3
```

```
>> M=W.*(W>0)
```

```
M =
```

```
21 0 0 12 5 10  
0 18 0 7 9 11  
17 0 0 8 13 6  
0 14 19 3 0 1  
16 0 20 0 0 2  
0 22 15 0 4 0
```

```
>> save zajecia2
```

```
>> t1='zielnona' {definiowanie łańcuchów}
```

```
t1 =  
zielnona  
>> t2='góra'  
t2 =  
góra  
>> t12=[t1 t2] {łączenie łańcuchów}  
t12 =  
zielnonagóra  
  
>> t12=[t1 t2]  
t12 =  
zielnonagóra  
>> t1='zielnona '  
t1 =  
zielnona  
>> t12=[t1 t2]  
t12 =  
zielnona góra  
>> t1='zielona '  
t1 =  
zielona  
>> t12=[t1 t2]  
t12 =  
zielona góra  
  
>> z=input('wpisz tekst ') {czeka na liczbę}  
wpisz tekst 325
```

z =

325

>> 325

ans =

325

>> z=input('wpisz tekst ')
wpisz tekst 87995545

z =

87995545

>> z=input('wpisz tekst ','s') {czeka na tekst}
wpisz tekst jjfjf

z =

jjfjf