

Wprowadzenie do środowiska Matlab

Zadanie 1: Utwórz zmienną $x=9$. Sprawdź jej wartość, a następnie oblicz pierwiastek kwadratowy ze zmiennej x .

Rozwiązanie:

```
1 % Zadanie 1
2 x = 9;
3 disp(x);
4 sqrt_x = sqrt(x);
5 disp(sqrt_x);
6
```

```
>> Zad_01
9
3
>>
```

Zadanie 2: Znajdź funkcje Matlaba, które wykorzystujemy do zaokrąglania liczb. Następnie użyj odpowiedniej z nich, aby:

- a. Wynikiem działania $6/5$ była liczba 1
- b. Wynikiem działania $6/5$ była liczba 2
- c. Wynikiem działania $9/5$ była liczba 1
- d. Wynikiem działania $9/5$ była liczba 2
- e. Wynikiem działania $-6/5$ była liczba -1
- f. Wynikiem działania $-6/5$ była liczba -2
- g. Wynikiem działania $-9/5$ była liczba -1
- h. Wynikiem działania $-9/5$ była liczba -2

```
>> Zad_02
1
2
1
2
-1
-2
-1
-2
>> |
```

```
1 % help round
2 % help ceil
3 % help floor
4 % help fix
5
6 a = floor(6/5);
7 disp(a);
8
9 b = ceil(6/5);
10 disp(b);
11
12 c = floor(9/5);
13 disp(c);
14
15 d = ceil(9/5);
16 disp(d);
17
18 e = ceil(-6/5);
19 disp(e);
20
21 f = floor(-6/5);
22 disp(f);
23
24 g = ceil(-9/5);
25 disp(g);
26
27 h = floor(-9/5);
28 disp(h);
29
```

Zadanie 3: Oblicz wartość następujących wyrażeń (Zadeklaruj oraz przypisz dowolne wartości całkowite do zmiennych x, y oraz z.)

a. $(x * y) - 2z$

b. $\frac{2z}{y}$

c. $\frac{2z}{y} * x$

d. $\sqrt{|x - 1|} - \sqrt[3]{|y|}$

e. $\arctg(z) + e^{-(x+3)}$

f. $\frac{3+e^{(y+1)}}{|y-\tan(z)|}$

g. $1 + |y - x| + \frac{(y-x)}{2} + \frac{|y-z|^3}{3}$

Rozwiązanie:

```
x = 4;
y = 3;
z = 2;

a = (x * y) - 2 * z;
disp(['Równanie a: ', num2str(a)])

b = 2 * z / y;
disp(['Równanie b: ', num2str(b)])

c = (2 * z / y) * x;
disp(['Równanie c: ', num2str(c)])

d = sqrt(abs(x - 1)) - nthroot(abs(y), 3);
disp(['Równanie d: ', num2str(d)])

e = atan(z) + exp(-(x + 3));
disp(['Równanie e: ', num2str(e)])

f = 3 + exp(y + 1) / abs(y - tan(z));
disp(['Równanie f: ', num2str(f)])

g = 1 + abs(y - x) + (y - x)^2 + abs(y - z)^3 / 3;
disp(['Równanie g: ', num2str(g)])
```

```
>> Zad_03
Równanie a: 8
Równanie b: 1.3333
Równanie c: 5.3333
Równanie d: 0.2898
Równanie e: 1.1081
Równanie f: 13.5299
Równanie g: 3.3333
>>
```

Zadanie 4: Wygeneruj trzy wektory dziesięcioelementowe X, Y, Z o dowolnych wartościach, następnie:

- zapisz wektor X do pliku o nazwie wektorX
- zapisz sumę wektorów Y i Z do pliku o nazwie wektorYZ
- wyczyść przestrzeń roboczą ze zmiennych X, Y, Z
- wczytaj wartości zmiennej X z pliku o nazwie wektorX

Rozwiązanie:

```
X = rand(1, 10); % wektor X
Y = rand(1, 10); % wektor Y
Z = rand(1, 10); % wektor Z

% a) zapisz wektor X do pliku
save('wektorX.mat', 'X');
disp(['Wektor X = ', mat2str(X)]);

% b) zapisz sumę wektorów Y i Z do pliku
YZ = Y + Z;
save wektorYZ YZ
disp(['Wektor YZ = ', mat2str(YZ)]);

% c) wyczyść przestrzeń roboczą
clear X Y Z;

% d) wczytaj wektor X z pliku
load('wektorX.mat');
disp(X);
```

```
>> Zad_04
Wektor X = [0.547008892286345 0.296320805607773 0.744692807074156 0.1889550
Wektor YZ = [1.57370410116242 1.15432206126867 1.29837209068565 0.96868417
Columns 1 through 6

    0.5470    0.2963    0.7447    0.1890    0.6868    0.1835

Columns 7 through 10

    0.3685    0.6256    0.7802    0.0811

>>
```

Zadanie 5: Dla 100-elementowego wektora A oblicz:

- a. sumę elementów,
- b. średnią elementów,
- c. medianę,
- d. odchylenie standardowe,
- e. liczbę elementów.

Rozwiązanie:

```
A = rand(1, 100); % 100-elementowy wektor A

a = sum(A);
disp(a);

b = mean(A);
disp(b);

c = median(A);
disp(c);

d = std(A);
disp(d);

e = length(A);
disp(e);
```

```
>> Zad_05
48.7924

0.4879

0.4941

0.2903

100
>>
```

Zadanie 6: Wykorzystując funkcje rand wygeneruj macierz C o pięciu wierszach i dziesięciu kolumnach, której elementy będą losowymi liczbami

całkowitymi z przedziału -10 do 10, a następnie:

- odwrócić w macierzy C kolejność kolumn (tzn. pierwsza kolumna ma stać się ostatnią, druga przedostatnią itd.);
- zamienić wiersz pierwszy z trzecim;
- zamienić ze sobą kolumny: druga z czwartą, szóstą z ósmą oraz dziesiątą z pierwszą;
- usunąć kolumny: piątą, szóstą i dziewiątą.

Rozwiązanie:

```
C = randi([-10, 10], 5, 10); % macierz C 5x10
disp('Macierz początkowa:');
disp(C);

% a) odwrócenie kolejności kolumn
C = fliplr(C);
disp('Macierz po odwróceniu kolejności kolumn:');
disp(C);

% b) zamiana wiersza pierwszego z trzecim
C([1, 3], :) = C([3, 1], :);
disp('Macierz po zamianie pierwszego i trzeciego wiersza:');
disp(C);

% c) zamiana kolumn
C(:, [2, 4]) = C(:, [4, 2]);
C(:, [6, 8]) = C(:, [8, 6]);
C(:, [10, 1]) = C(:, [1, 10]);
disp('Macierz po zamianie kolumn:');
disp(C);

% d) usunięcie kolumn 5, 6 i 9
C(:, [5, 6, 9]) = [];
disp('Macierz po usunięciu kolumn 5, 6 i 9:');
disp(C);
```

Macierz po zamianie kolumn:
Columns 1 through 9

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

Column 10

-3
8
8
4
-6

Macierz po usunięciu kolumn 5, 6 i 9:

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

```
>> Zad_06
Macierz początkowa:
Columns 1 through 9

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

Column 10

8
8
-3
4
-6

Macierz po odwróceniu kolejności kolumn:
Columns 1 through 9

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

Column 10

2
-5
2
4
-6
```

Macierz po zamianie pierwszego i trzeciego wiersza:
Columns 1 through 9

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

Column 10

2
-5
2
4
-6

Zadanie 7: Napisać skrypt Matlab, który zmiennej *a* przypisze wartość 11.
Wywołać ten skrypt i sprawdzić poprawność wyniku

Rozwiązanie:

```
% Skrypt przypisujący wartość 11 do zmiennej a  
a = 11;  
  
% Wyświetlenie wartości zmiennej a  
disp(['Wartość zmiennej a: ', num2str(a)]);
```

```
>> Zad_07  
Wartość zmiennej a: 11  
>>
```

Zadanie 8: Utwórz macierz $A = \begin{bmatrix} 1 & 2 & 3 \\ -4 & -5 & -6 \\ 7 & 8 & 9 \end{bmatrix}$

- wyznacz kolejno sumy: po każdym wierszu, po każdej kolumnie i po dwóch przekątnych,
- wygeneruj macierz B o rozmiarze 3x3 wartości pseudolosowych z zakresu -10 do 10,
- wyznacz najmniejszą wartość z całej macierzy B,
- wykonaj działania: $A + B$, $A - B$, $A \cdot B$,
- usuń pierwszy wiersz z macierzy A oraz drugą kolumnę z macierzy B,
- obróć macierze otrzymane w podpunkcie e o 90 stopni

Rozwiązanie:

```
A = [1 2 3; -4 -5 -6; 7 8 9]; % macierz A
disp('Początkowa macierz A:');
disp(A);

% a) sumy
suma_wierszy = sum(A, 2);
disp('Suma po każdym wierszu:');
disp(suma_wierszy);

suma_kolumn = sum(A, 1);
disp('Suma po każdej kolumnie:');
disp(suma_kolumn);

przekatna1 = sum(diag(A)); % suma pierwszej przekątnej
disp('Suma pierwszej przekątnej:');
disp(przekatna1);

przekatna2 = sum(diag(flipud(A))); % suma drugiej przekątnej
disp('Suma drugiej przekątnej:');
disp(przekatna2);

% b) macierz B o wartościach pseudolosowych
B = randi([-10, 10], 3, 3);
disp('Macierz B o wartościach pseudolosowych:');
disp(B);

% c) najmniejsza wartość z macierzy B
min_B = min(B(:));
disp('Najmniejsza wartość z macierzy B:');
disp(min_B);

% d) działania na macierzach
suma_macierzy = A + B;
disp('Suma macierzy A i B:');
disp(suma_macierzy);

roznica_macierzy = A - B;
disp('Różnica macierzy A i B:');
disp(roznica_macierzy);

mnozenie_macierzy = A .* B;
disp('Mnożenie macierzy A i B (element po elemencie):');
disp(mnozenie_macierzy);

% e) usuwanie wiersza i kolumny
A(1, :) = [];
disp('Macierz A po usunięciu pierwszego wiersza:');
disp(A);

B(:, 2) = [];
disp('Macierz B po usunięciu drugiej kolumny:');
disp(B);
```

```
% f) obrót macierzy o 90 stopni
A_rot = rot90(A);
disp('Macierz A po obrocie o 90 stopni:');
disp(A_rot);

B_rot = rot90(B);
disp('Macierz B po obrocie o 90 stopni:');
disp(B_rot);
```

```
>> Zad_08
Początkowa macierz A:
     1     2     3
    -4    -5    -6
     7     8     9

Suma po każdym wierszu:
     6
    -15
    24

Suma po każdej kolumnie:
     4     5     6

Suma pierwszej przekątnej:
     5

Suma drugiej przekątnej:
     5

Macierz B o wartościach pseudolosowych:
     2     8    -7
    -7    -10    10
    -5     0     4

Najmniejsza wartość z macierzy B:
    -10

Suma macierzy A i B:
     3    10    -4
    -11   -15     4
     2     8    13

Różnica macierzy A i B:
    -1    -6    10
     3     5   -16
    12     8     5

Mnożenie macierzy A i B (element po elemencie):
     2    16   -21
    28    50   -60
   -35     0    36

Macierz A po usunięciu pierwszego wiersza:
    -4    -5    -6
     7     8     9

Macierz B po usunięciu drugiej kolumny:
     2    -7
    -7    10
    -5     4
```

```
Macierz A po obrocie o 90 stopni:
    -6     9
    -5     8
    -4     7

Macierz B po obrocie o 90 stopni:
    -7    10     4
     2    -7    -5
>>
```

Zadanie 9: Dla zdefiniowanej macierzy $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

- wyświetl jej drugi wiersz,
- wyświetl jej trzecią kolumnę,
- wyświetl jej pierwszy wiersz i czwartą kolumnę, wynik przypisz zmiennej B.

Rozwiązanie:

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

% a) drugi wiersz
second_row = A(2, :);
disp(second_row);

% b) trzecia kolumna
third_column = A(:, 3);
disp(third_column);

% c) pierwszy wiersz i czwarta kolumna
B = [A(1, :) A(:, 4)'];
disp(B);
```

```
>> Zad_09
4     5     6

3
6
9

1     2     3     3     6     9

>> Zad_09
4     5     6

3
6
9

Index in position 2 exceeds array bounds. Index must not exceed 3.

Error in Zad_09 (line 12)
B = [A(1, :) A(:, 4)'];
      ^^^^^^^
>>
```

Zadanie 10: 10. Napisać funkcję o nazwie **zwiększac** przyjmującą jeden argument x i zwracającą argument powiększony o $x * 2 + 10$

Rozwiązanie:

```
function result = zwiększac(x)
    result = x * 2 + 10;
end
```

```
a = 7;
result = zwiększac(a);
disp(['Wynik funkcji zwiększac dla x = ', num2str(a), ': ', num2str(result)]);
```

```
>> Zad_10
Wynik funkcji zwiększac dla x = 7: 24
>>
```


Zadanie 11: Napisz funkcję sprawdzającą, czy dana liczba jest parzysta.

Rozwiązanie:

```
function wynik = czy_parzysta(x)
    % Sprawdza, czy liczba x jest parzysta
    if mod(x, 2) == 0
        wynik = true;
    else
        wynik = false;
    end
end
```

```
liczba = 7;
if czy_parzysta(liczba)
    disp([num2str(liczba), ' jest parzysta.']);
else
    disp([num2str(liczba), ' jest nieparzysta.']);
end
```

```
>> Zad_11
8 jest parzysta.
>> Zad_11
7 jest nieparzysta.
>>
```

Zadanie 12: Napisz instrukcję switch wyświetlającą na podstawie zmiennej całkowitej nr nazwę miesiąca słownie.

Rozwiązanie:

```
nr = 1;

switch nr
    case 1
        disp('Styczeń');
    case 2
        disp('Luty');
    case 3
        disp('Marzec');
    case 4
        disp('Kwiecień');
    case 5
        disp('Maj');
    case 6
        disp('Czerwiec');
    case 7
        disp('Lipiec');
    case 8
        disp('Sierpień');
    case 9
        disp('Wrzesień');
    case 10
        disp('Październik');
    case 11
        disp('Listopad');
    case 12
        disp('Grudzień');
    otherwise
        disp('Nieprawidłowy numer miesiąca');
end
```

```
>> Zad_12
Kwiecień
>> Zad_12
Maj
>> Zad_12
Nieprawidłowy numer miesiąca
>> Zad_12
Styczeń
>>
```

Zadanie 13: Napisz instrukcję, która liczy sumę n początkowych liczb nieparzystych.

Rozwiązanie:

```
n = 5;
suma = 0;

for i = 1:2:(2*n - 1) |
    suma = suma + i;
end

disp(['Suma ', num2str(n), ' początkowych liczb nieparzystych to: ', num2str(suma)]);
```

```
>> Zad_13
Suma 5 początkowych liczb nieparzystych to: 25
>>
```

Zadanie 14: Napisz skrypt, który będzie generował losowo liczbę całkowitą z przedziału $[-1000, 1000]$ i będzie wyprowadzał informację, czy liczba jest parzysta czy nieparzysta.

Rozwiązanie:

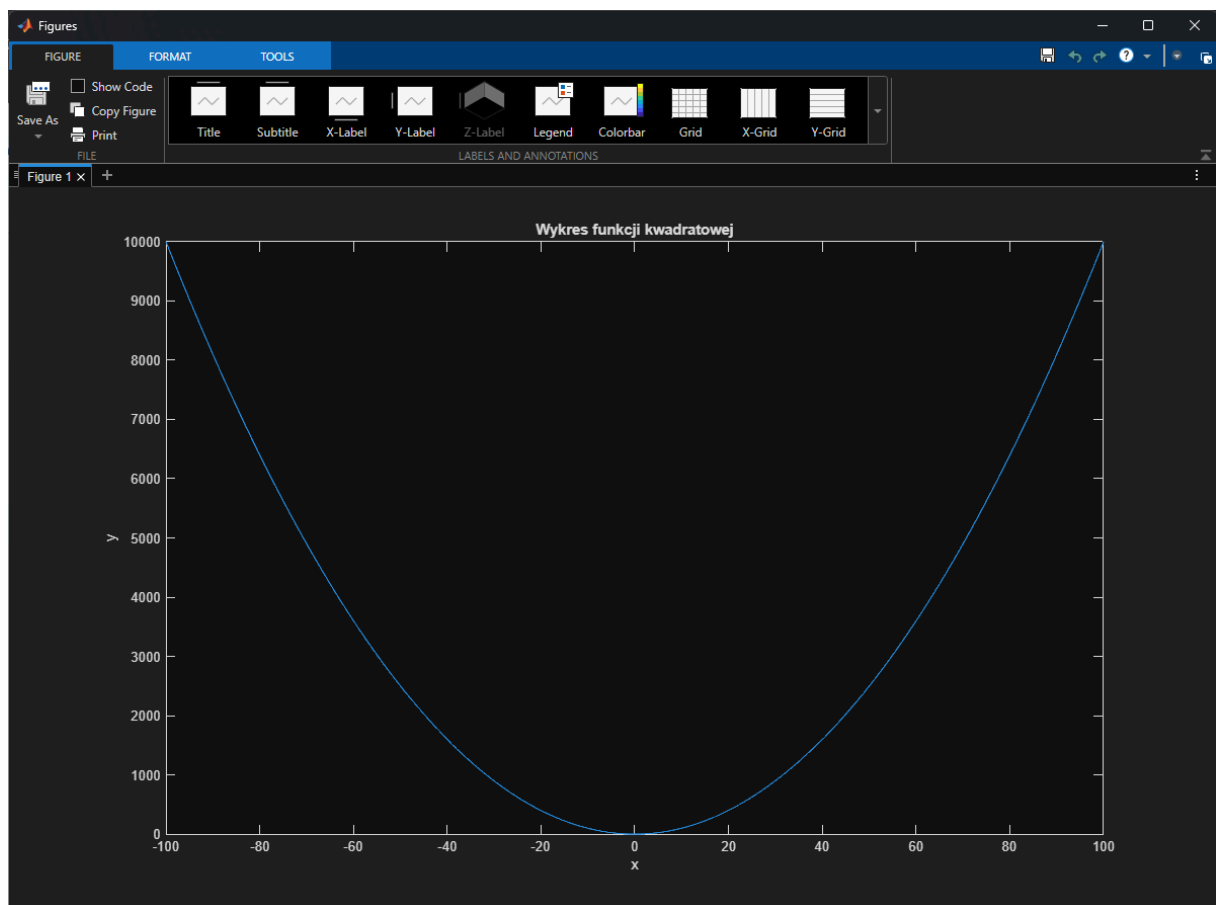
```
random_number = randi([-1000, 1000]);
if mod(random_number, 2) == 0
    disp('Liczba parzysta');
else
    disp('Liczba nieparzysta');
end
disp(num2str(random_number));|
```

```
>> Zad_14
Liczba parzysta
812
>> Zad_14
Liczba parzysta
-746
>> Zad_14
Liczba nieparzysta
827
>> Zad_14
Liczba nieparzysta
265
>>
```

Zadanie 15: Narysuj wykres dowolnej funkcji kwadratowej w przedziale $[-100, 100]$

Rozwiązanie:

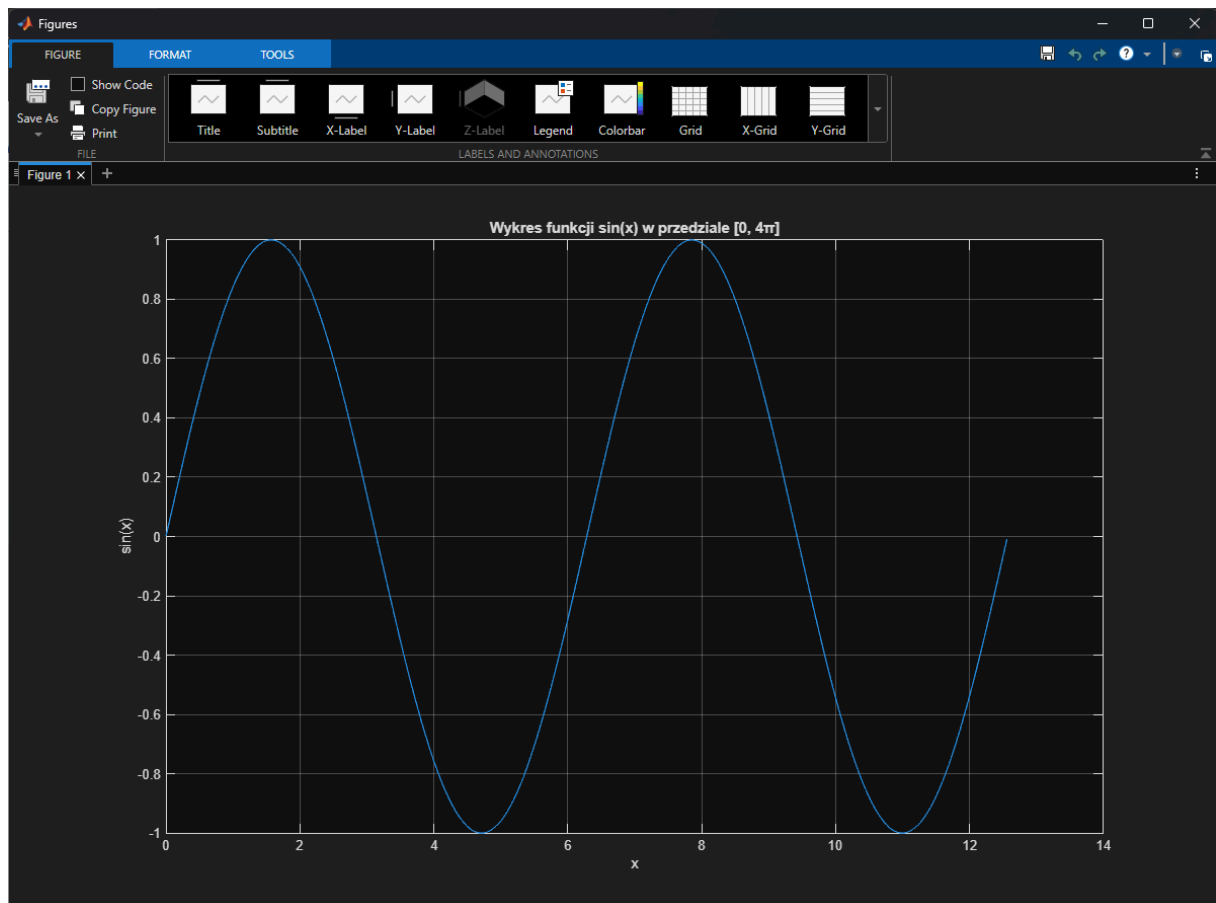
```
x = -100:0.1:100;  
y = x.^2;  
plot(x, y);  
title('Wykres funkcji kwadratowej');  
xlabel('x');  
ylabel('y');
```



Zadanie 16: Dla x z przedziału od 0 do 4π narysuj przebieg funkcji $\sin(x)$

Rozwiązanie:

```
x = 0:0.01:4*pi;  
y = sin(x);  
  
plot(x, y);  
title('Wykres funkcji sin(x) w przedziale [0, 4π]');  
xlabel('x');  
ylabel('sin(x)');  
grid on;
```



Zadanie 17: Napisanie skryptu, który tworzy dwie losowe macierze o takich samych wymiarach, a następnie dodaje je do siebie i wyświetla wyniki.

Rozwiązanie:

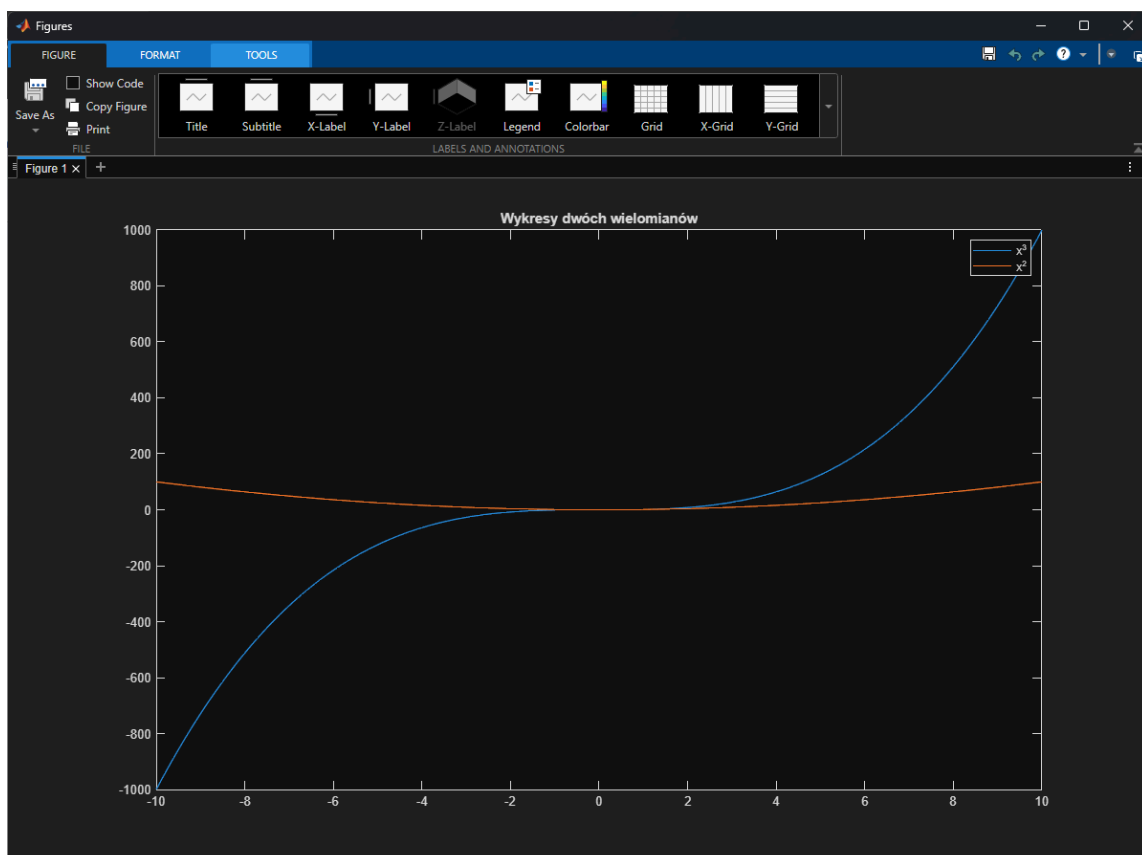
```
A = rand(3, 3);  
B = rand(3, 3);  
C = A + B;  
disp(C);
```

```
>> Zad_17  
    0.8978    1.8732    1.6263  
    0.4204    1.7571    0.9929  
    0.9686    1.1171    1.3345  
  
>>
```

Zadanie 18: W jednym układzie współrzędnych narysuj wykresy dwóch dowolnych wielomianów.

Rozwiązanie:

```
x = -10:0.1:10;  
y1 = x.^3;  
y2 = x.^2;  
plot(x, y1, x, y2);  
title('Wykresy dwóch wielomianów');  
legend('x^3', 'x^2');
```



Zadanie 19: Opracuj skrypt, który dla dwóch zmiennych wejściowych oblicza: kwadrat pierwszego argumentu, sumę wszystkich argumentów, iloczyn wszystkich argumentów.

Rozwiązanie:

```
function [kwadrat1, suma, iloczyn] = obliczenia(x, y)|
    kwadrat1 = x^2;
    suma = x + y;
    iloczyn = x * y;
end
```

```
a = 5;
b = 3;

[k1, s, il] = obliczenia(a, b);

disp(['Kwadrat pierwszego argumentu: ', num2str(k1)]);
disp(['Suma argumentów: ', num2str(s)]);
disp(['Iloczyn argumentów: ', num2str(il)]);
|
```

```
>> Zad_19
Kwadrat pierwszego argumentu: 25
Suma argumentów: 8
Iloczyn argumentów: 15
>>
```

Zadanie 20: Napisz program, który po podanych przez użytkownika wymiarach macierzy generuje zadaną macierz o losowych elementach. Następnie dla wygenerowanej macierzy oblicza:

- Sumę elementów na przekątnej
- Sumę elementów każdego wiersza
- Sumę elementów każdej kolumny
- Minimalny i maksymalny element macierzy

Rozwiązanie:

```
clear;

wiersze = input('Podaj liczbę wierszy: ');
kolumny = input('Podaj liczbę kolumn: ');

wiersze = floor(wiersze);
kolumny = floor(kolumny);

M = randi([-100, 100], wiersze, kolumny);
disp('Wygenerowana macierz:');
disp(M);

min_wymiar = floor(min([wiersze, kolumny]));
suma_przekatnej = sum(diag(M(1:min_wymiar, 1:min_wymiar)));
disp(['Suma elementów na przekątnej: ', num2str(suma_przekatnej)]);

suma_wierszy = sum(M, 2);
disp('Suma elementów wierszy:');
disp(suma_wierszy);

suma_kolumn = sum(M, 1);
disp('Suma elementów kolumn:');
disp(suma_kolumn);

min_el = min(M(:));
max_el = max(M(:));
disp(['Minimalny element macierzy: ', num2str(min_el)]);
disp(['Maksymalny element macierzy: ', num2str(max_el)]);
```

```
>> Zad_20
Podaj liczbę wierszy:
5
Podaj liczbę kolumn:
7
Wygenerowana macierz:
    83    -85    55   -48   -47    10     3
   -70   -12    64    60   -71   -71   -20
    65   -79    74   -14   -73    71   -85
     8    93   -84    83    74    25   -52
   100  -100   -20   -64    16   -30   -76

Suma elementów na przekątnej: 244
Suma elementów wierszy:
   -29|
  -120
   -41
   147
  -174

Suma elementów kolumn:
   186  -183    89    17  -101     5  -230

Minimalny element macierzy: -100
Maksymalny element macierzy: 100
>>
```