```
% MATLAB, which stands for Matrix Laboratory, is a program
%        for performing numerical and symbolic calculations. It is widely used
%        in science and engineering, as well as in mathematics.
% The MathWorks, Inc. 3 Apple Hill Drive, Natick, MA 01760, www.mathworks.com
cos(pi/5)                  % result sent to the variable ans
format rat                 % returns rational expressions
x=5.4                      % sends a number to the variable x
format compact             % eliminates the extra spacing

format long
  %  format short    4 decimal places (with or without e) - DEFAULT
  %  format long     15 decimal places (with or without e)
disp('- - - - - - - - - - - - - - format long')

clear  % name1 name2 use clear for your convenience
       % removes all (or specified) variables from workspace

s = 1 -1/2 + 1/3 ...       % continuation
    - 1/4 + 1/5 - 1/6
i, j^2                     % two commands in one line
y=sqrt(-x);                % do not display the result
disp(y)                    % displays a variable on the screen
L = input('liczba = ');    % prompt and waiting for input from keyboard

disp('- - - - - - - - - - - - - special variables')
disp([' eps =     ',num2str(eps)])        % precision
disp([' realmin = ',num2str(realmin),',   realmax = ',num2str(realmax)])
disp([' 1/0 -->   ',num2str(Inf), ',          0/0 -->   ',num2str(NaN)])

% Legal names consist of any combination of letters and digits,
% starting with a letter.  Do not use: pi, eps, i, j, Inf, NaN. realmin, realmax
% Variable names are case sensitive, so X and x are not equal.
%
% To avoid having to retype long expressions use the up arrow key  to scroll through
% lines previously typed. Typing one or more characters and then the up arrow key
% displays previous lines that begin with those characters.
%
% Built-In Functions:
%   sqrt, exp, log, log10, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh
%   abs, sign, conj, imag, real, angle, round, rem

% EXERCISE:  use 1,i,pi,e in an expression yielding 0

% System commands:  pwd, ls, dir, type, date, cputime, lookfor
disp('- - - - - - - - - - - - - Help')
help rem, lookfor TXT,  playshow intro %  see also Help box in MATLAB desktop

disp('- - - - - - - - - - - - - Matrices and Vectors')
A=[5 3 1; 6 9 4; 1 2 8]            % ; separates rows
A(2,3), A(1,:), A(:,2), A(:,[1,3]),     % selected row, column, columns, entry
A',          E=inv(A)              % transposition, inversion
c=[1; 2; 3];
x=A\c          %  solve system of equations Ax=c,  also over- & under- determined

disp('- - - - - - - - - - - - - eigenvalues ')
[P,L]=eig(A)
cond(A), det(A)
spectrum=diag(L);
sort(spectrum)

disp('- - - - - - - - - - - - - selected operations')
a=[1:3]                    % collon notation
a(2), a(1,2), a(1:2:3)
b=[10:2:14]
B=[a;b]
B(:,1)=[]
l=length(b)
[m,n]=size(B)

cross(a,b)
dot(a,b), sum(a.*b)        % .* - componentwise operation
norm(a)                    %  Euklidean for vector
norm(A)                    %  Spectral radius (maximum singular value in general)
                           %  other: norm(X,1), norm(X,inf), norm(X,'fro')
d=diag([21,22,23,24,25])+eye(5)          % diagonal and unit matrices

% Built-In Functions
%   length, size, dot, cross, eye, diag (create or extract diagonals)
%   cond, norm, rank, det, trace, chol, \, inv, eig, svd, min, max
% EXERCISE:  a) solve a system of linear equations, b)verify the result,
%            c) solve the equations again, using the inverse matrix,
%            d)evaluate eigenvalues and conditioning number,
%            e)evaluate norms of the matrix, its inverse and the solution,
%            f) evaluate the angle between two vectors (1,2,3), (-1,2,3)    (answer: ~31deg)
%
%>>> disp('- - - - - - - - - - - - - Symbolic operations')
%>>> solve('sin(x)+x=5')
%>>> solve('a*x^2+b*x+c','x')
%>>> S=solve('x+y=a','x+2*y=b'),    S.x,    S.y          % data structure
%>>> syms r d f
%>>> x = r*cos(d)*cos(f);        y = r*cos(d)*sin(f);         z = r*sin(d);
%>> J = jacobian([x; y; z], [r d f]),  detJ=det(J),   detJ=simple(detJ)   → -cos(d)*r^2
%>> v=subs(detJ,{r,d,f},{3,2,1}),      double(v)                → -9*cos(2) →3.7453
%>> f=sin(x)*cosh(x)*y,            diff(f,x,2),                  int(f,x)
```

```
disp('- - - - - - - - - - - - - - Plots')
t=0:0.01:pi;              tt = linspace(-pi,pi,25);      % 25 points between -π and π
x=sin(20*t).*exp(-pi*t);          y=cos(5*t).*exp(-t);
plot(t,x,'k','linewidth',2), hold on,     plot(t,y,'g--','linewidth',2), hold off

axis([0,3,-1,1]),  grid,  xlabel('\alpha'),     ylabel('\beta'),      title('t e x t')
%  subplot(m,n,k)
save test.dat  tt  -ASCII  -DOUBLE,          pp = load('test.dat')
Q=t'*t;    contourf(Q),        bar([1,2,3,4,5],[1,3,5,4,2])
disp('- - - - - - - - - - - - - - Control flow')
n=123;
if rem(n,2)==0 & n>5     % IF              %  <=  ||  ~
   b=1;
 else                                  EXAMPLES
   b=0;                                figure(2);   hold on;   axis equal
end                                    axis([-1 1 -1 1]); axis off
                                       N=100;  FI=3*pi;
s=0;                                   for n=1:N
for m=1:2:n              % FOR           r= [1/N*(n-1)     1/N*n];
   s=s+m;                              fi=[FI/N*(n-1)   FI/N*n];
end                                    x=r.*cos(fi);    y=r.*sin(fi);
                                       plot(x,y,'c','linewidth',3);
ds=0.1;                                pause(0.01)
while abs(ds)>1e-5       % WHILE      end
   ds=ds/5;                            hold off
   s=s+ds;                             %%%%%%%%%%%%%%%%%%%
   if s>1e5
     break               % BREAK       function [s]=silnia(n)
   end                                 % evaluates  n!=1*2*3*…*n
end                                    s=1;
                                       for k=1:n
switch rem(n,4)          % SWITCH      s=s*k;
   case {0,1}                          end
     M = n
   case {2,3}
     M = 2*n
   otherwise
     error('This is impossible')
end
disp('- - - - - - - - - - - - - - Delaunay triangulation')     %%%%%%%%%%%%
xy=[0 0; 1 0;  1 1;  0 1;  0.2 0.4;   0.8 0.6];
x = xy(:,1);      y = xy(:,2);      TRI = delaunay(x,y)
figure(3)
triplot(TRI,x,y,'y','linewidth',3),      hold on;
voronoi(x,y,TRI,'r')
plot(x,y,'oc','linewidth',3),     hold off;     axis equal;      axis([0,1,0,1])
```

```
% iloczyn skalarny
a=[1:3], b=[10:2:14]
dot(a,b)
sum(a.*b)

% kąt miedzy wektorami
phi=acos(dot(a,b)/( norm(a)*norm(b) ))
% iloczyn wektorowy
% norma
nw=norm(a)*norm(b)*sin(phi)
nw1=norm(cross(a,b))
c=cross(a,b)
c1=cross(b,a)
dot(a,c)
dot(b,c)

% transformacja wektora
phi=pi/2
T=[cos(phi) -sin(phi) 0; sin(phi) cos(phi) 0; 0 0 1]
b1=T*b'
norm(b), norm(b1)

% gradient
v = -2:0.2:2;
[x,y] = meshgrid(v);
z = x .* exp(-x.^2 - y.^2);
[px,py] = gradient(z);
contour(v,v,z)
hold on
quiver(x,y,px,py)

% tensor
a=rand(3)
a1=T*a*T'
det(a)
det(a1)

% wykresy
surf(x,y,z)
c=contourf(x,y,z)
clabel(c)
colorbar
```