1.

a) function [R1, R2, varargout] = fun(varargin)

b) function [varargout] = fun(x1, x2, varargin)

c) function [N, err, X] = funX, Y)

d) function y = fun(x)

2.

a) Треба ввести у командне вікно команду help і потім назву нашої функції, в данному випадку myfun:

help myfun

b) За допомогою функції type, тоді це буде мати вигляд:

type myfun

c) Використовується команда lookfor ну і наша назва функції:

lookfor myfun

3.

a) function x = myfun(F,D,x0,tol,maxIter)

b) за допомогою дескриптора @ (анонімна функція)  
myfun=@(x1,x2,...,xn)(Expr(x1,x2,...,xn))

c) за допомогою команди inline (inline-функція)  
myfun=inline(‘Expr(x1,x2,...,xn)’);

function pl7\_3a % зовнішня функція  
clear all, close all, clc  
f1 = @(x)(x.\*x); % анонімна функція  
f2 = inline('2.\*x'); % inline-функція

4.

Після опису основної функції (саме вона і тільки вона доступна ззовні),  
можуть бути розташовані підфункції, опис яких починається ключовим словом  
function, а закінченням тіла є кінець файла або останній оператор перед  
початком опису нової підфункції або ключове слово end. Виклик підфункцій можливий тільки з основної функції або підфункцій, розміщених у даному файлі. В описі основної функції можуть бути розташовані (в довільному місці) вкладені  
функції, опис яких починається ключовим словом function, а закінчення тіла  
обов’язково фіксується ключовим словом end. Використовувати (поєднувати) обидва ці варіанти в одній М-функції допускається тільки при умові, що останнім оператором основної функції і підфункцій обов’язково є end.

5.

a)

MATLAB має можливості використовувати імена функцій як аргументи  
(процедурний тип даних). При описі функції такі імена записуються звичайним  
способом, а при виклику – до фактичного імені додається дескриптор (@) або  
записується ім’я у вигляді текстової константи, наприклад, опис деякої функції f:  
function [...] = f(...,fun,...)  
Тоді e=Expr(fun(...),...);  
– виклик функції fun (...) у складі виразу Expr ;  
E=Expr1(f(...,@myfun,...),...);  
або  
E=Expr1(f(...,'myfun',...),...);  
– виклик функції f у складі виразу Expr1, де в якості фактичного параметра, який  
відповідає fun, використовуємо ім’я myfun.

b)

function x = newton(F,D,x0,tol,maxIter) %файл з функцією

switch nargin

case [1,2]

warning('Недостатня кількість аргументів')

x = NaN;

return

case 3

tol = 1e-4; maxIter = 1e4;

case 4

maxIter = 1e4;

end

for i = 1:maxIter

y = D(x0)\F(x0);

x = x0-y;

if norm(x-x0)<tol

return

end

x0 = x;

end

warning('Метод розбігається.')

x = NaN;

end

clear %основний файл же викликається дана функція

clc

p = (0:0.05:2)\*pi; x1 = sqrt(3.8)\*cos(p); x2 = sqrt(15.3)\*sin(p);

plot(x1,x2,'r','LineWidth',1.5)

hold on

x2 = -3:0.05:3; x2 = x1.^2+2;

plot(x1,x2,'b','LineWidth',1.5)

xmin = -3; xmax = 3; ymin = 0; ymax = 6;

plot([xmin xmax],[0 0],'k','LineWidth',1.5)

plot([0 0],[ymin ymax],'k','LineWidth',1.5)

axis image

axis([xmin xmax ymin ymax])

grid on

grid minor

legend('x1.^2/3.8 + x2.^2/15.3 = 1',' x2 - x1.^2 = 2 ')

F = @(x) [ (x(1)^2)/3.8+(x(2)^2)/15.3-1; ...

x(2) - x(1)^2 - 2 ];

D = @(x) [2.\*x(1)/3.8, 2.\*x(2)/15.3; ...

-2.\*x(1), 1];

disp('Початкове наближення х0 - вектор довжини 2: ')

x0 = input('x0 = ');

s = size(x0);

if s(1) == 1

x0 = x0';

end

x = newton(F,D,x0,1e-5,1e2); % викликається наша функція newton

if ~isnan(x)

disp('Розв\*язок системи x =')

disp(x')

plot(x(1),x(2),'ko','MarkerSize',7,'MarkerFaceColor','g', ...

'DisplayName','Розв''язок')

else

disp('ERROR')

end

hold off