```
%problem 3
tol = 1e-9;

%a
f = @(x) ((x).^3).*log(x) - 3.*(x);
df = @(x) (3.*(x).^2).*log(x) + ((x).^2) -3;
dff = @(x) (6.*(x)).*log(x) + 5.*(x);

%newton
tic
[f_newt, iterf_newt] = newton(df,dff,1.5,tol);
toc

tic
[f_gold, iterf_gold] = golden_search(f,1,1.5,tol);
toc


%b
g = @(x) -1*( -(x(1) ^2) /x(2) + x(1) *log( -x(1) ) - x(2) ^2) ;
dg = @(x) [ -2*x(1)/x(2) + log(-x(1)) + 1;
    ( x(1).^2)/x(2).^2 - 2*x(2)];
dgg = @(x) [1/x(1) + -2/x(2)  , 2*x(1)/(x(2).^2) ;
    2*x(1)/(x(2).^2) , -2*(x (1).^2) /(x (2).^3) - 2];
%newton (cant start with [-1,1]?)
tic
[g_newt, iterg_newt] = newton2 (dg ,dgg ,[ -0.1;0.1] ,tol);
toc

%fminsearch
tic
[g_min,fval , exitflag , output ] = fminsearch (g ,[ -0.1;0.1]);
toc


%c
h = @(x) x.^6 - 0.75.*x.^5 + 1.5.*x.^4 + 1.25.*x.^3 - 3.*x.^2;
dh = @(x) 6.*x.^5 - 3.75.*x.^4 + 6.*x.^3 + 3.75.*x.^2 - 6.* x;
dhh = @(x) 30.*x.^4 - 15.*x.^3 + 18.*x.^2 + 7.5.*x -6;

tic
[h_l,fval , exitflag , output ] = fminsearch (h ,-1);
toc
```

```
%[h_back, opt, iterh_back] = grad_descent_backtracking(h,dh,1,tol,0.5,0.8)
tic
[h_ex, opt, iterh_ex] = grad_descent_exact(h,dh,-5,tol);
toc

%d
l = @(x) (x(1) ^2 -2) ^2 + (log(x(2) ) -2) ^2 + (x(3) +x(2) ) ^2;

%fmin search
tic
[x_l,fval , exitflag , output ] = fminsearch (l ,[0;1;0])
toc
```

```
%p6
%declare objective function and derivatives
f = @(x) (x')*c*x;
df = @(x) 2*x*c;
dff = @(x) 2*c;

%covariance matrix
c = [106.991461 23.892217 -27.843578 -59.063559 6.484768 0.007298
23.892217 93.627323 -117.56474 -40.45979 0.502487 0.006454
-27.843578 -117.56474 226.216329 16.106375 6.320048 -0.006226
-59.063559 -40.45979 16.106375 60.962192 -6.718355 -0.006654
6.484768 0.502487 6.320048 -6.718355 1.234519 -0.0009
0.007298 0.006454 -0.006226 -0.006654 -0.0009 0.000013];

%construct A matrix
A = [0.2083, 0.1874, 0.2111, 0.1095, 0.0239,0.001
    1, 1, 1, 1, 1, 1];

%construct b matrix
b = [0.12; 1];

%initialize starting weight matrix
w = zeros(6,1);

%use newton for equality constrained problems
[x,opt ,iter ,nu] = newton_eq (f,df ,dff ,w ,A, b ,1e-9)
```