```matlab
%
% ============================================================================
% 2.086 - Numerical Computation for Mechanical Engineers
% Fall 2016
% ============================================================================
% Last Edited By: S. Sroka
% Date          : 8/27/16
%
% ------------------------
% Assignment 1
% Problem    1
% Points     25
%
%
% Workspace Variables:
% N = an integer between 1 and 100 inclusive
%
% Output Variables:
% Q = sum from n = 1 to n = N of n^(-2)
%

Q = 0;
for n = 1:N
    Q = Q + n^-2;
end


%
% ============================================================================
% 2.086 - Numerical Computation for Mechanical Engineers
% Fall 2016
% ============================================================================
% Last Edited By: S. Sroka
% Date          : 8/27/16
%
% ------------------------
% Assignment 1
% Problem    2
% Points     25
%
%
% Workspace Variables:
% Nfib = an integer between 1 and 25 inclusive
%
% Output Variables:
% Ffib = The Nth element of the Fibonacci Sequence

if Nfib == 1
    Ffib    = 1;
else
    Ffib_2  = 0;
    Ffib_1  = 1;
    for n = 1:Nfib-1;
        Ffib   = Ffib_1 + Ffib_2;
        Ffib_2 = Ffib_1;
        Ffib_1 = Ffib;
    end
end
```

```matlab
%
% ==========================================================================
% 2.086 - Numerical Computation for Mechanical Engineers
% Fall 2016
% ==========================================================================
% Last Edited By: S. Sroka
% Date          : 9/5/16
%
% ------------------------
% Assignment 1
% Problem    3
% Points     25
%
% Hardcoded variables:
% x_bar = the fixed point in the Taylor series
%         FUN FACT: A Taylor series expanded about 0 (i.e. with x_bar = 0)
%         is called a Maclaurin series.
% The function f(x) = sin(x), and therefore the first derivative is cos(x),
% the second derivative is -sin(x), and the third is -cos(x).
%
% Workspace Variables:
% n = the index of the highest Taylor term between 1 and 8 inclusive
% x = the location at which we evaluate the function and its derivatives between 0 and pi
%     inclusive
%
% Output Variables:
% P = finite Taylor series approximation of the function at the location x
%

%% Part a
x_bar = 0;
P = 0;
x_tilde = (x-x_bar);
for k = 0:n
    if (k==0) || (k==4) || (k==8)
        P = P + sin(x_bar)*x_tilde^k/factorial(k);
    elseif (k==1) || (k==5)
        P = P + cos(x_bar)*x_tilde^k/factorial(k);
    elseif (k==2) || (k==6)
        P = P + -sin(x_bar)*x_tilde^k/factorial(k);
    elseif (k==3) || (k==7)
        P = P + -cos(x_bar)*x_tilde^k/factorial(k);
    end
end

%% Part b
comment3b='For a given x_tilde, the finite Taylor series that includes more terms (i.e.
has a greater value of n) will better approximate the true function.';

%% Part c
n = 1;
x = pi/4;
x_bar = 0;
% the maximum value of the abosolute value of f^(2) = -sin(x) on the
% interval [0 pi/4] is sin(pi/4) = 0.707
max_sin = abs(-sin(x));
x_tilde = x-x_bar;
max_err = max_sin*x_tilde^2/factorial(n+1);
comment3c='The error epsilon is -0.078291, the absolute value of which is smaller than
the maximum error of 0.218090. Epsilon can be reduced by reducing x_tilde (i.e. the grid
spacing) or if the next derivative (omitted from the finite Taylor series) decreases. A
means to achieve the second epsilon reduction mechanism is to increase the order of the
derivative in the error bound (i.e. increase n). Full credit is awarded for any of the
aforementioned answers.';
```

```
%
% ==========================================================================
% 2.086 - Numerical Computation for Mechanical Engineers
% Fall 2016
% ==========================================================================
% Last Edited By: S. Sroka
% Date          : 8/27/16
%
% ------------------------
% Assignment 1
% Problem    4
% Points     25
%
%
% Hardcoded Variables
% f(z) = atan((0.5*z).^3)+0.02*log(z+10)+0.02*z
% df/dz = 1./(1+(0.5*z).^6).*3.*(0.5.*z).^2.*0.5+0.02.*(1./(z+10))+0.02
%
% Workspace Variables:
% zInitial = the initial guess to begin Newton's method -3 <= z^* <= 3
% kMax     = the last index of the solution to Newton's method
%            (equivalently there are kMax iterations) 2 <= kMax <= 100
%
% Output Variables:
% zFinal   = the solution using Newton's method for kMax iterations
%
% Note in an mfile the third-fifth lines would be on one line.

z_1 = zInitial;
for k = 2:kMax
    z_star = z_1 -
(atan((0.5*z_1).^3)+0.02*log(z_1+10)+0.02*z_1)/(1./(1+(0.5*z_1).^6).*3.*(0.5.*z_1).^2.*0.
5+0.02.*(1./(z_1+10))+0.02);
    z_1 = z_star;
end
zFinal =z_1;
```