```
% This is a good idea if you are going to run this script many times
clear ;

% This is the system of units we are using

M = 1. ;
K = 1. ;
x0 = 1. ;
w0 = sqrt(K/M) ;


% We will take the initial velocity to be zero
% This is the initial velocity in units of  x0*w0
% i.e. this is vbar = v/ (x0*w0)
v0 = 0. ;

% This is the value of the damping constant
% in our unit system, i.e. this is bbar = b/(m *w0)
B  = 0.22;

% This defines the times step in units of w0
dt = 0.001 ;
% This defines the final time and initial time
tmin =0. ;
tmax =10.;
% This defines an array t(0) = tmin , t(1) = tmin + dt, t(2) =tmin + 2dt,
% t(3) = tmin + 3*dt, ...
t = [tmin:dt:tmax]  ;

% We will store our solution in these arrays
% x(1) =0 , x(2) = 0, x(3) = 0...
% Later we will fill these arrays up with the actual solution
x = zeros(1, length(t) ) ;
v = zeros(1, length(t) ) ;

% This array contains the work done by friction from 0...t
Wfr = zeros(1, length(t) ) ;

% Set the initial value of xn
% and the initial value of vn
% and the initial time  tn
xn = x0 ;
vn = v0 ;
tn = tmin ;

% Store this inital value of x and the initial value of velocity
x(1) = xn;
v(1) = vn;

% We will use this to determine when the velocity crosses zero
vold = vn ;
```

```
W = 0. ;

for i = 2:length(t)

    F = -K*xn - B*vn ; % Computes the force

    dW = -B*vn*vn*dt ; % Computes the work done by friction

    % This step determines the n+1 values of xn, vn, tn
    xn=xn+ vn*dt ;
    vn=vn + F/M*dt ;
    tn=tn + dt;

    W = W + dW ;

    % Check if the velocity crosses zero. If it crosses zero we print out
    % the value In the case of matlab, we can print out a variable simply
    % by typing the variable name.
    if vold*vn < 0
        tn
    end
    vold = vn ;

    % Store the values of the  xn in the appropriate time slot.
    x(i) = xn ;
    v(i) = vn ;
    Wfr(i) = W ;

end

% Given the array v(t) we compute the kinetic energy as an
% array, we also compute the potential energy as an array.
% After this line KE(1) = 0.5*M*v(1)^2 , KE(2) = 0.5*M*v(2)^2.
% Similarly we have PE(1) = 0.5*K*x(1)^2, PE(2) = 0.5*K*x(2)^2
KE = 0.5*M*v.^2  ;
PE = 0.5*K*x.^2  ;

Etot = KE + PE - Wfr ;
% print out the initial energy
Etot(1)
% print out the final energy
Etot(length(t))


% This shows how to plot a function of the array t
% after this line
% xsol(1) = x0*cos(w0*t(1)), xsol(2)=x0*cos(w0*t(2)), ....
xsol = cos(w0*t) ;

% Uncomment this line to plot the  x versus t ,
% and the zero damping solution x= x0*cos(w0*t)
% plot(t, x, t, xsol) ;
```

```
% Uncomment this line to plot the Kinetic, Potential and Total Energies
% versus time. Here the energies are in units of Mw0^2 x0^2=  k x0^2
plot(t, KE, t, PE, t, -Wfr, t, Etot) ;
```