1. Modify Lec2D so that it also prints the message "integer roots" if \( r_1 \) and \( r_2 \) are real and each is within \( 10^{-6} \) of an integer.

```matlab
% Lec2D
% Solves ax^2 + bx + c = 0
clc
a = input('Enter a: ');
b = input('Enter b: ');
c = input('Enter c: ');
d = b^2 -4*a*c;
if d >= 0
    r1 = (-b-sqrt(d))/(2*a);
    r2 = (-b+sqrt(d))/(2*a);
    fprintf('Root1 = %10.6f
', r1)
    fprintf('Root2 = %10.6f
', r2)
else
    realPart = -b/(2*a);
    imagPart = sqrt(-d)/(2*a);
    fprintf('Root1 = %10.6f+i*%10.6f
', realPart, imagPart)
    fprintf('Root2 = %10.6f-i*%10.6f
', realPart, imagPart)
end
```

2. Modify Lec2E so that it plots one hundred random points within the square. (Use a for-loop).

```matlab
% Lec2E
% Generates 5 random points within the square
% with vertices (0,0), (1,0), (1,1), (0,1).

% Set up the window
close all
figure
hold on
axis equal
axis([0 1 0 1])

% Plot the points
plot(rand,rand,'*r','markersize',20)
plot(rand,rand,'*b','markersize',20)
plot(rand,rand,'*g','markersize',20)
plot(rand,rand,'*m','markersize',20)
plot(rand,rand,'*k','markersize',20)

% Select your favorite
[x,y] = ginput(1);
plot(x,y,'o','markersize',25)
```

3. Modify Lec3B so that it prints the value \( \cos(\theta/2^j) \) for \( j = 10, 11, \ldots, 20 \). Exploit the half-angle formula and use a for-loop.

```matlab
% Lec3B
% Repeated application of the "half angle" formula
% \[
% \cos(z/2) = \sqrt{ (1 + \cos(z))/2 } 
% \]
clc
deg = input('Enter an angle in degrees between 0 and 180: ');
theta = pi*deg/180;

% Method 1. Compute \( \cos(\theta/2^j) \) via half-angle formula...
```
\[ x = \cos(\theta); \]
\[ x = \sqrt{(1+x)/2}; \quad \cos(\theta/2) \]
\[ x = \sqrt{(1+x)/2}; \quad \cos(\theta/4) \]
\[ x = \sqrt{(1+x)/2}; \quad \cos(\theta/8) \]
\[ \text{fprintf}(\'\text{Method 1: } \cos(\%6.3f/8) = \%20.16f\', \text{deg}, x) \]

% Method 2. Compute \( \cos(\theta/8) \) directly...
\[ y = \cos(\theta/8); \]
\[ \text{fprintf}(\'\text{Method 2: } \cos(\%6.3f/8) = \%20.16f\', \text{deg}, y) \]
\[ \text{error} = \text{abs}(x-y); \]
\[ \text{fprintf}(\'\text{Absolute Error = } %5.3e\', \text{error}) \]

4. Modify \texttt{Lec3D} so that it properly prints out the ending times should they be of the form "x minutes after 12".

% \texttt{Lec3D}
% When does this lecture end?
% Clear the command window and get time information
\texttt{clc}
\texttt{S = input(\'When does the lecture start? hhmm = \');}
\texttt{L = input(\'How long does the lecture last? hhmm = \');}

% Isolate the hours and minutes
\texttt{S_hr = floor(S/100);}
\texttt{S_min = mod(S,100);}
\texttt{L_hr = floor(L/100);}
\texttt{L_min = mod(L,100);}

% Add the parts
\texttt{End_min = S_min + L_min;}
\texttt{End_hr = S_hr + L_hr;}

% Adjust minute value
\texttt{if End_min > 59}
\texttt{    End_min = mod(End_min,60);}
\texttt{    End_hr = End_hr + 1;}
\texttt{end}

5. Let \( f_j \) be the \( j \)th Fibonacci number. Modify \texttt{Lec4D} so that it prints \( f_1, \ldots, f_{20} \) and then all the integers \( k \) that satisfy \( f_{20} < k < f_{21} \).

% \texttt{Lec4D}
% Fibonacci Numbers and Golden Ratio
\texttt{clc}
\texttt{disp(\' j j-th Fibonacci# j-th Ratio\')}
\texttt{disp(\'-------------------------------------------\')}
\texttt{N = 12;}
\texttt{x = 0;}
\texttt{y = 1;}
\texttt{for j=1:N}
\texttt{    \% x is the (j-2)nd Fibonacci number}
\texttt{    \% y is the (j-1)st Fibonacci number}
\texttt{    \% z is the j-th Fibonacci number}
\texttt{    z = x + y;}
\texttt{    r = z/y;}
\texttt{    fprintf(\'\%3d \%10d \%20.6f\n\',j,z,r)\}
\texttt{x = y;}
\texttt{y = z;}
\texttt{end}
\texttt{close all}
\texttt{fill([0 r r 0],[0 0 1 1 0],\'m\')}
\texttt{title(\'The Perfect Rectangle: Length/Width = ( 1 + sqrt(5) )/2\',\'fontsize\',24)}
\texttt{axis equal off}