In this assignment, you are asked to write a Matlab program which fits an arbitrary set of points 
\[(x_k, y_k), \ k = 1, \ldots, m\]
in the least squares sense by a polynomial of given degree, represented in the form
\[y(x) = c_1 \phi_1(x) + \cdots + c_n \phi_n(x)\].

You will then apply your program to a particular set of data fitting problems, and see how the 
choice of basis functions impacts the accuracy of your results. Note: Matlab uses double precision 
and gives a maximum of approximately 16 decimal digits of precision.

Suggested guidelines for program

Write a Matlab function, \(function [c, \text{conditionNo}] = \text{leastSquares}(x, y, n, \text{basis})\), say, which uses 
either of the following two sets of basis functions:

(A) the “monomial” basis: \(\phi_j(x) = x^{j-1} \ j = 1, \ldots, n\),

(B) the “Legendre polynomial” basis: \(\phi_j(x) = L_j \left(-1 + 2 \frac{x-a}{b-a}\right) \ j = 1, \ldots, n\),
where \([a, b] = [\min x_k, \max x_k]\),
depending on the value of \(\text{basis}\). More about choice (B) below...

In either case, the corresponding normal equations \(R^T R c = R^T y\) are formed \((R_{i,j} = \phi_j(x_i), 1 \leq i \leq m, 1 \leq j \leq n)\), then Gaussian elimination with partial pivoting is applied via the following 
Matlab commands for solving \(Ac = b\):
\[
\begin{align*}
[L, U, P] &= lu(A); \\
b &= P \ast b; \\
y &= L \backslash b; \\
c &= U \backslash y;
\end{align*}
\]
The condition number of the coefficient matrix is also computed via Matlab command \(\text{cond}(A)\).

The Legendre polynomials \(\{L_j(t)\}_{j=1}^\infty\) are defined by the conditions:
\[
\begin{align*}
&i) \ L_j(t) \text{ is a polynomial of degree } j - 1 \\
&ii) \ L_j(1) = 1 \ \text{ for all } j \\
&iii) \int_{-1}^{1} L_i(t) L_j(t) \, dt = 0 \ \text{ if } i \neq j
\end{align*}
\]
and are of special interest over the interval \(t \in [-1, 1]\). The first few are:
\[
L_1(t) = 1, \quad L_2(t) = t, \quad L_3(t) = \frac{1}{2}(3t^2 - 1), \quad L_4(t) = \frac{1}{2}(5t^3 - 3t),
\]
as pictured on the next page. Given any two consecutive Legendre polynomials, the next one in 
the family may be gotten via the 3-term recurrence relation:
\[
L_j(t) = \frac{2j-3}{j-1} t L_{j-1}(t) - \frac{j-2}{j-1} L_{j-2}(t), \ j = 3, 4, \ldots, n.
\]
In order to make the Legendre polynomials appropriate as basis functions for least squares approximation over \( x \in [a, b] \), we map \( x \in [a, b] \) into \( t \in [-1, 1] \) via the transformation \( t = -1 + 2 \frac{x-a}{b-a} \), as indicated in (B) above.

**Approximation problems for you to solve**

**I.** Data: \((0, 1); (1, 1); (2, 1); (3, 4); (4, 3)\)

Approximations: polynomials of degree 1, 2, 3, 4.

Use only the first set of basis functions. For each approximation, print the corresponding computed coefficients and condition number. (You can do this using 'fprintf' or by simply omitting semicolons after appropriate statements. If you choose the latter, use the statement 'format compact' to save space.) Plot your four approximations in a \( 2 \times 2 \) pattern on a single page using the Matlab command 'subplot' (type 'help subplot' for information). Include the five data points on each plot. Note: We used these data points for our example in class.

**II.** Data: \( x = [0:0.2:1]' \), \( y = x + 1 \) (51 data points)

Approximations: polynomials of degree 4, 8, 12 \((n = 5, 9, 13)\).

Use both sets of basis functions, A and B. For each approximation problem (6 in all), print the computed coefficients, condition number, and the relative error in your computed coefficients, i.e.,

\[
\frac{\|c_{\text{computed}} - c_{\text{exact}}\|}{\|c_{\text{exact}}\|}.
\]

Be sure to display all the digits in your computed coefficients. If you’re printing via omission of semicolons, use the statement 'format long' to make this happen. Note: Normally, it’s not possible for least squares approximations to give an exact fit to all the data points. But here an exact fit is possible, and that is what must happen, apart from computer roundoff error. This should tell you what \( c_{\text{exact}} \) is for these least squares problems.

Provide a brief interpretation of your results. Is the condition number a good indicator of the accuracy of your computed solutions? Which set of basis functions is better?