Topic: Linear interpolation, polynomial interpolation, spline interpolation
Read: Chapter 17 intro, sections 17.1,4,5 Chapter 18 sections 1,2,5 & 6
Create a plot for every problem unless otherwise instructed. All plots should include data points (use a marker) and curve fit (use a solid line; with different colors & legend when more than one curve fit). Label and title all plots!

Handwork problem:

HW7_1 For the following data

x	–4	–2	0	1	2	3
y	–10	3	8	25	52	36

a) Compute the linear spline equation for each segment and simplify to get the slope-intercept form. Include the domain.
b) Interpolate at x = -3.5, 0.8 and 2.3.

Coding problems:

HW7_2 This problem uses an interpolating polynomial to estimate the area under a curve. Fit the interpolating polynomial to the following set of points. These points are the actual values of $f(x) = \sin(e^x - 2)$

x	0	0.4	0.8	1.2	1.6
y	–0.8415	–0.4866	0.2236	0.9687	0.1874

a) Plot the function f(x) and the interpolating polynomial, using different colors. Use `polyfit` and `polyval`. Also include the data points using discrete point plotting.
b) We wish to estimate the area under the curve, but this function is difficult to integrate. Hence, instead of finding $\int_0^{1.6} \sin(e^x - 2)\,dx$ (which is the same as finding the area under the curve $\sin(e^x - 2)$), we will compute the area under the interpolating polynomial over the domain 0 < x < 1.6. Polynomials are easy to integrate. So, after doing the integration by hand, use MATLAB to compute the value of the exact integral. Print the result using `fprintf`.
c) Do you believe that this is a good estimate? Write in comments.

HW7_3 Ball bearings are hardened through a process known as quenching—submersion of the heated ball bearing in oil or water in order to cool it rapidly. The data below represent the temperature of the ball at various points in the cooling process

time (sec)	0	30	60	150	240
Temp (C)	800	457	269	79	46

Plot the data (as discrete points). In the same figure window, also plot

a) interpolating polynomial, (use `polyfit` and `polyval`)

b) piecewise cubic interpolation (not-a-knot) and

c) piecewise cubic interpolation (shape-preserving)

Include a legend for the three methods used: polynomial interpolation, not-a-knot, shape-preserving.

Then looking at the plot, choose the method you believe to be the best and smoothest fit of all these to predict the temperature after 3 minutes of cooling. Print the result using `fprintf`.

HW7_4

The acceleration due to gravity at an altitude *y* above the surface of the earth is given by

y, m	0	30,000	60,000	90,000	120,000
g, m/s^2	9.8100	9.7487	9.6879	9.6278	9.5682

Compute g at *y*=55,000 m. Determine the answer using both linear and polynomial interpolation. Additionally, find the altitude at which the gravitational constant is 9.75 m/s^2, using linear interpolation. Print the answers to the screen using `fprintf`.

HW7_5

For the same data given for 7_1,

x	–4	–2	0	1	2	3
y	–10	3	8	25	52	36

a) Plot the data points with a circle marker (MATLAB).

b) Plot the unique interpolating polynomial (passes exactly through each point) and print the equation of the polynomial (in the form $y = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0$) to the screen using `fprintf`. Don't use `polyfit` & `polyval` for this problem; use the backslash on the Van der Monde matrix & construct the function handle and use `fplot`.

c) Plot the linear splines between each pair of points using `interp1`.

d) Plot the "not-a-knot" cubic spline using `interp1`