General Information: This assignment contains written and/or programming tasks. Combine all the answers to the written tasks in a single PDF document, named `{lastname}-written.pdf`. You can also scan or take pictures of (readable) handwritten papers. JPEG/PNG image files are accepted in this case and they should be named `{exercisenumber}-{lastname}-written.{jpeg/png}`. Make sure that we can follow the manual calculations. Do not combine too many small steps into one. The programming tasks have to be solved in *Julia* and the source code files have to be submitted using the following naming scheme: `{exercisenumber}-{lastname}.jl`.

(1) (2.5 Points) Perform ordinary least squares (OLS) on the California housing dataset (california_housing.csv) using the variables median_income and median_house_value. The file ols.jl contains preprocessing steps necessary to select data corresponding to these values, simplify, reshape, and split the data into a training and test set.
We will use the train set to conduct OLS, and check one approximation for novel data via the test set. To do so, we will use one of the obtained approximations to predict the median_house_value values for the test set.

a) (0.5 Points) Conduct **linear** OLS approximation using the preprocessed training set in Julia. Follow the instructions outlined in ols.jl, plot the resulting polynomial (w.r.t. the training data points), and calculate the approximation error. Does the approximation describe the data sufficiently?

b) (0.5 Points) Analogously to the prior task, perform **quadratic** OLS, plot the polynomial, and calculate the approximation error. Does the fitted curve capture the data better than its linear counterpart?
Hint: We are looking for the coefficients a_0, a_1, and a_2 of $p(x) = a_0 + a_1x + a_2x^2$ that minimize the approximation error $\sum_{j=1}^{n} |a_2x_j^2 + a_1x_j + a_0 - y_j|^2$.

c) (0.5 Points) Using **Cramer's rule**, conduct **cubic** OLS, plot the polynomial, and calculate the approximation error. Compare the resulting approximation to its predecessors. You are allowed to use det() from the Linear Algebra package for Cramer's rule.
Hint: Similar to quadratic OLS, we are looking for the coefficients a_0, a_1, a_2, and a_3 of $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ that minimize the approximation error $\sum_{j=1}^{n} |a_3x_j^3 + a_2x_j^2 + a_1x_j + a_0 - y_j|^2$. Note that this is linked to the Vandermonde matrix.

d) (0.5 Points) Use the **test set** for inference and check the generalizability of (one of) your approximations. To this end, plot the actual data points and project the dependent variable median_income onto one of the previously obtained curves. Calculate the approximation error (on the test set w.r.t. the selected approximation) and reason whether or not the approximation generalizes to unseen data.

e) (0.5 Points) Briefly describe the concepts and consequences of overfitting and underfitting generally first, and then in this context. Without explicitly trying other polynomials, which order could lead to overfitting/underfitting, and which order would you deem a good fit?

(2) (2.5 Points) Apply (general) barycentric interpolation to the triangle defined in barycentric_interpolation.jl. Follow the outlined instructions, and don't use any routines readily available in Julia. You may use some functions e.g. cross(), norm(), sign() from the Linear Algebra package. Save the resulting image and submit it as barycentric_interpolation_result.png together with your other files. Your triangle should feature similar colors to the one depicted below.

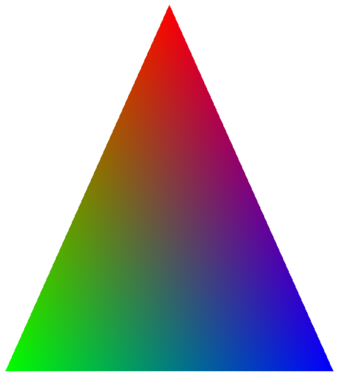

Figure 1: Triangle color interpolation

(3) (2.0 Points)

a) (0.5 Points) Determine the 3rd order polynomial $p(x)$ such that

$$\begin{aligned} p(-1) &= 0 \\ p'(-1) &= -5 \\ p(1) &= 10 \\ p'(1) &= 20 \end{aligned}$$

using Hermite interpolation.

b) (0.5 Points) Construct the Lagrange interpolating polynomial $p(x)$ for the function $f(x) = \sin(\ln(x))$ using the nodes $x_0 = 2$, $x_1 = 2.75$, $x_2 = 4$ and determine the error form for this polynomial (as given in the lecture slides).

c) (1.0 Points) Prove that bilinear interpolation is separable. Specifically, show that the final formula for bilinear interpolation in a unit square:

$$f(x, y) = \begin{pmatrix} 1-x \\ x \end{pmatrix}^T \begin{pmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{pmatrix} \begin{pmatrix} 1-y \\ y \end{pmatrix}$$

can be derived by first performing a linear interpolation along the x-axis, then the y-axis. Further, prove that the opposite holds by showing that we get the same result if we interpolate along the y-axis first, and then the x-axis.