1. **Finite-Element Method**
Implement the Finite-Element Method as discussed in the lecture and use it to solve the Poisson equation,

$$\frac{\mathrm{d}^2 U(x)}{\mathrm{d}x^2} = -4\pi\rho(x)\,, \quad 0 < x < 1\,, \tag{1}$$

for $\rho(x) = 1/(4\pi)$ with the boundary conditions $U(0) = 0$ and $U(1) = 1$.

(a) Solve the equation for different values of $N \in [3, 100]$, where N is the number of elements. Plot the results and compare them to the analytic solution $U(x) = -x(x-3)/2$.

(b) When the number of nodes is large enough, their positions should not significantly influence the result. Change your program such that it can use a user specified list of nodes, which do not have to be equidistant. Repeat part (a) with randomly chosen nodes.

(c) Change your program such that it can be used for x-dependent charge densities. Now the calculation of $b_i = \int_{x_i}^{x_{i+1}} dx\, 4\pi\rho(x)\phi_i$ in the system of linear equations requires numerical integration. You can either implement one of the methods you learned in this course, use an external library[1], or explicitly provide a list of precomputed integral values[2].

(d) Use your program to solve the Poisson equation for $\rho(x) = x/(4\pi)$ using the same boundary conditions as above. Also solve it analytically and compare the result to the result obtained with the finite-element method.

[1] In Python, you can use the `scipy.integrate` module.
[2] You can use for example Mathematica's `NIntegrate` function to obtain them.