```
# Import dependencies
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange
```

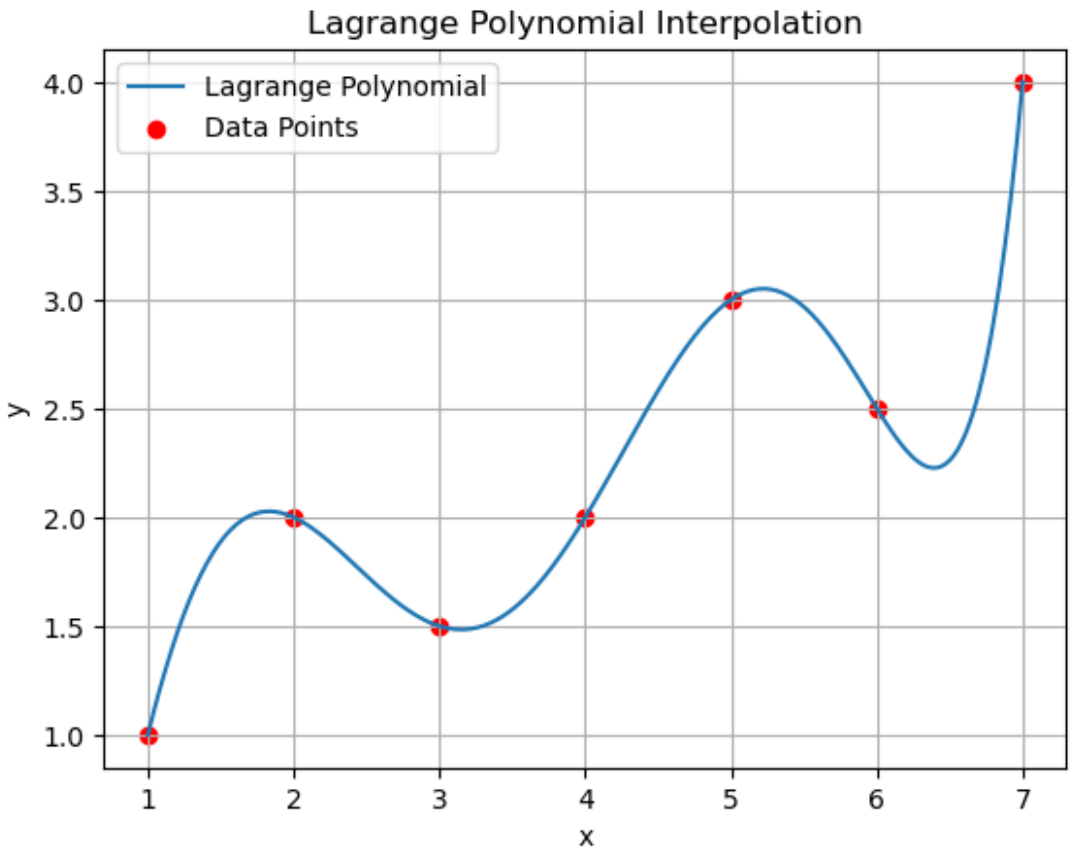
Problem 1.1

```
# Define x and y
x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([1, 2, 1.5, 2, 3, 2.5, 4])

# Perform Lagrange interpolation
polynomial = lagrange(x, y)

# Generate x-axis points for plotting the interpolated polynomial
x_plot = np.linspace(1, 7, 300)
y_plot = polynomial(x_plot)

# Plot the interpolated polynomial and data points
plt.plot(x_plot, y_plot, label="Lagrange Polynomial")
plt.scatter(x, y, color='red', label="Data Points")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Lagrange Polynomial Interpolation')
plt.legend()
plt.grid(True)
plt.show()
```

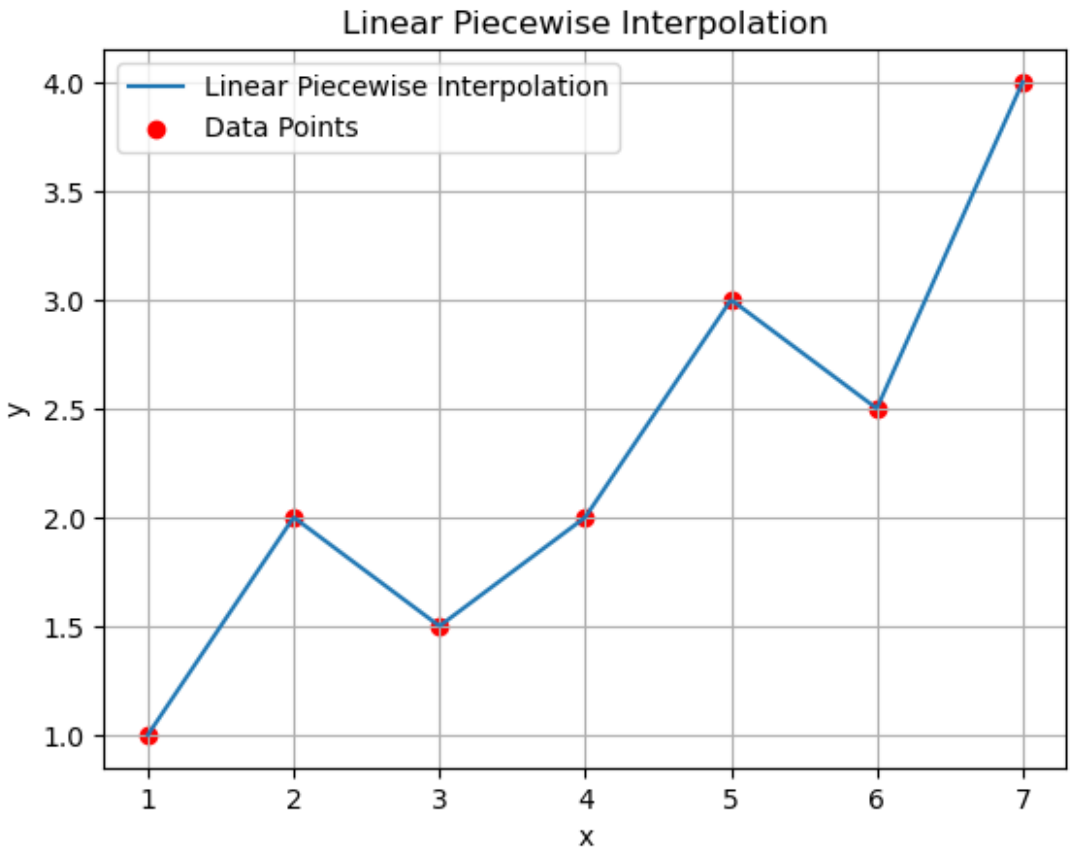


Problem 1.2

```
# Define x and y values
x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([1, 2, 1.5, 2, 3, 2.5, 4])

# Perform linear piecewise interpolation
x_plot = np.linspace(1, 7, 300)
y_plot = np.interp(x_plot, x, y)

# Plot the linear interpolation and data points
plt.plot(x_plot, y_plot, label="Linear Piecewise Interpolation")
plt.scatter(x, y, color='red', label="Data Points")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Piecewise Interpolation')
plt.legend()
plt.grid(True)
plt.show()
```



Problem 1.3

```
# Define x and y values
x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([1, 2, 1.5, 2, 3, 2.5, 4])

# Fit a polynomial of degree 6 (n=6)
coeffs_n6 = np.polyfit(x, y, 6)
poly_n6 = np.poly1d(coeffs_n6)

# Fit a polynomial of degree 3 (n=3)
coeffs_n3 = np.polyfit(x, y, 3)
poly_n3 = np.poly1d(coeffs_n3)

# Generate x-axis points for plotting
x_plot = np.linspace(1, 7, 300)
y_n6 = poly_n6(x_plot)
y_n3 = poly_n3(x_plot)

# Plot both degree 6 and degree 3 polynomials along with data points
plt.plot(x_plot, y_n6, label="Degree 6 Polynomial", color='blue')
plt.plot(x_plot, y_n3, label="Degree 3 Polynomial", color='green')
plt.scatter(x, y, color='red', label="Data Points")
plt.xlabel('x')
```

```
plt.ylabel('y')
plt.title('Least Squares Polynomial Fit (n=3 and n=6)')
plt.legend()
plt.grid(True)
plt.show()
```