```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

def linear_fit(x, y):
    """
    Implement linear least squares fitting
    Returns slope (m) and intercept (c)
    """
    n = len(x)
    sum_x = np.sum(x)
    sum_y = np.sum(y)
    sum_xy = np.sum(x * y)
    sum_x2 = np.sum(x ** 2)

    m = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x ** 2)
    c = (sum_y - m * sum_x) / n

    return m, c

def plot_fit(x, y, y_fit, title):
    """
    Helper function to plot data points and fitted curve
    """
    plt.figure(figsize=(10, 6))
    plt.scatter(x, y, color='blue', label='Data points')
    plt.plot(x, y_fit, 'r-', label='Fitted curve')
    plt.xlabel('x')
```

```
    plt.ylabel('y')
    plt.title(title)
    plt.legend()
    plt.grid(True)
    plt.show()

def forward_difference_table(x, y):
    """
    Compute forward difference table
    """
    n = len(y)
    table = np.zeros((n, n))
    table[:,0] = y

    for j in range(1, n):
        for i in range(n-j):
            table[i,j] = table[i+1,j-1] - table[i,j-1]

    return table

def backward_difference_table(x, y):
    """
    Compute backward difference table
    """
    n = len(y)
    table = np.zeros((n, n))
    table[:,0] = y
```

```
    for j in range(1, n):
        for i in range(j, n):
            table[i,j] = table[i,j-1] - table[i-1,j-1]

    return table

# Task 1: Linear Curve Fitting
print("Task 1: Linear Curve Fitting")
x1 = np.array([1, 2, 3, 4, 5])
y1 = np.array([3, 6, 8, 11, 15])

m, c = linear_fit(x1, y1)
y1_fit = m * x1 + c

print(f"Slope (m): {m:.4f}")
print(f"Intercept (c): {c:.4f}")
plot_fit(x1, y1, y1_fit, "Linear Curve Fitting")

# Task 2: Polynomial Curve Fitting
print("\nTask 2: Polynomial Curve Fitting")
x2 = np.array([0, 1, 2, 3, 4])
y2 = np.array([2, 3, 6, 11, 18])

coeffs = np.polyfit(x2, y2, 2)
y2_fit = np.polyval(coeffs, x2)

print(f"Coefficients (a, b, c): {coeffs[0]:.4f}, {coeffs[1]:.4f}, {coeffs[2]:.4f}")
plot_fit(x2, y2, y2_fit, "Polynomial Curve Fitting")
```

```
# Task 3: Exponential Curve Fitting
print("\nTask 3: Exponential Curve Fitting")
x3 = np.array([1, 2, 3, 4, 5])
y3 = np.array([2.5, 4.7, 8.8, 16.2, 30.3])

def exp_func(x, a, b):
    return a * np.exp(b * x)

popt, _ = curve_fit(exp_func, x3, y3)
y3_fit = exp_func(x3, *popt)

print(f"Coefficients (a, b): {popt[0]:.4f}, {popt[1]:.4f}")
plot_fit(x3, y3, y3_fit, "Exponential Curve Fitting")

# Task 4: Three Constants Model
print("\nTask 4: Three Constants Model")
x4 = np.array([1, 2, 3, 4, 5])
y4 = np.array([2.1, 3.6, 6.3, 11.5, 18.9])

def custom_func(x, a, b, c):
    return a + b * np.log(x) + c * x**2

popt, _ = curve_fit(custom_func, x4, y4)
y4_fit = custom_func(x4, *popt)

print(f"Coefficients (a, b, c): {popt[0]:.4f}, {popt[1]:.4f}, {popt[2]:.4f}")
plot_fit(x4, y4, y4_fit, "Three Constants Model Fitting")
```

```
# Task 5: Forward Difference Table
print("\nTask 5: Forward Difference Table")
x5 = np.array([0, 1, 2, 3, 4])
y5 = np.array([1, 3, 7, 13, 21])

forward_table = forward_difference_table(x5, y5)
print("\nForward Difference Table:")
print("x\ty\tΔ¹y\tΔ²y\tΔ³y\tΔ⁴y")
for i in range(len(x5)):
  print(f"{x5[i]}", end="\t")
  for j in range(min(i+1, 5)):
    print(f"{forward_table[i,j]:.0f}", end="\t")
  print()

# Task 6: Backward Difference Table
print("\nTask 6: Backward Difference Table")
x6 = np.array([5, 6, 7, 8, 9])
y6 = np.array([1, 8, 27, 64, 125])

backward_table = backward_difference_table(x6, y6)
print("\nBackward Difference Table:")
print("x\ty\t∇¹y\t∇²y\t∇³y\t∇⁴y")
for i in range(len(x6)):
  print(f"{x6[i]}", end="\t")
  for j in range(min(i+1, 5)):
    print(f"{backward_table[i,j]:.0f}", end="\t")
  print()
```

```
# Task 7: Higher-Order Differences
print("\nTask 7: Higher-Order Differences")
x7 = np.array([0, 1, 2, 3, 4])
y7 = np.array([1, 8, 27, 64, 125])

forward_table_7 = forward_difference_table(x7, y7)
print("\nForward Difference Table:")
print("x\ty\tΔ¹y\tΔ²y\tΔ³y\tΔ⁴y")
for i in range(len(x7)):
  print(f"{x7[i]}", end="\t")
  for j in range(min(i+1, 5)):
    print(f"{forward_table_7[i,j]:.0f}", end="\t")
  print()
```