Explanation of Python Program

Aymen Negadi

1 Importing Libraries

```
import numpy as np
```

This line imports the NumPy library and assigns it the alias `np`. NumPy is used for numerical operations such as computing means, sums, and array manipulations.

```
import matplotlib.pyplot as plt
```

This line imports the Matplotlib plotting module and assigns it the alias `plt`. It is used to create graphs such as scatter plots and regression lines.

2 Function: `compute_regression`

```
def compute_regression(x, y):
```

This line defines a function named `compute_regression`. It takes two parameters:

- `x`: the input data (years of experience)
- `y`: the output data (salary)

```
x_mean = np.mean(x)
```

This line computes the average (mean) value of the input data `x`. It corresponds to the mathematical quantity $\bar{x}$.

```
y_mean = np.mean(y)
```

This line computes the average (mean) value of the output data `y`. It corresponds to the mathematical quantity $\bar{y}$.

```
a = np.sum((x - x_mean) * (y - y_mean)) / np.sum((x - x_mean)**2)
```

This line computes the slope a of the regression line using the Least Squares formula:

$$a = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

The slope represents how much the salary changes for one additional year of experience.

```
b = y_mean - a * x_mean
```

This line computes the intercept b of the regression line using the formula:

$$b = \bar{y} - a\bar{x}$$

The intercept represents the estimated salary when experience is zero.

```
return a, b
```

This line returns the computed slope and intercept to the caller of the function.

3 Function: predict

```
def predict(x, a, b):
```

This line defines a function that performs prediction using the regression model.

```
return a * x + b
```

This line applies the linear regression equation:

$$y = ax + b$$

It computes and returns the predicted salary for a given input value.

4 Function: plot_regression

```
def plot_regression(x, y, a, b):
```

This line defines a function responsible for plotting the data and the regression line.

```
y_pred = predict(x, a, b)
```

This line computes the predicted salary values for all input data points. These values are used to draw the regression line.

```
plt.scatter(x, y, label="Real data")
```

This line plots the original dataset as scattered points. Each point represents one employee.

```
plt.plot(x, y_pred, label="regression line")
```

This line draws the regression line using the predicted values.

```
plt.xlabel("years of experience")
```

This line labels the horizontal axis of the graph.

```
plt.ylabel("Salary (in thousands of DA)")
```

This line labels the vertical axis of the graph.

```
plt.title("linear regression (manual calcul using functions)")
```

This line sets the title of the graph.

```
plt.legend()
```

This line displays a legend to distinguish between real data points and the regression line.

```
plt.show()
```

This line displays the graph window.

5 Main Program

```
x = np.array([1, 2, 3, 4, 5], dtype=float)
```

This line creates a NumPy array containing years of experience. The data type is set to `float` to allow numerical computations.

```
y = np.array([30, 35, 45, 50, 60], dtype=float)
```

This line creates a NumPy array containing the corresponding salaries.

```
a, b = compute_regression(x, y)
```

This line calls the regression function and stores the slope and intercept.

```
print("a (slope) =", a)
```

This line prints the value of the slope.

```
print("b (intercept) =", b)
```

This line prints the value of the intercept.

```
print("Equation :  y = a*x + b")
```

This line prints the mathematical form of the regression model.

6 Prediction

```
x_test = 3
```

This line defines a test input value representing three years of experience.

```
print("salary predicted for 3 years =", predict(x_test, a, b))
```

This line predicts and prints the salary for three years of experience.

7 Graph Display

```
plot_regression(x, y, a, b)
```

This line calls the plotting function to display the final regression graph.