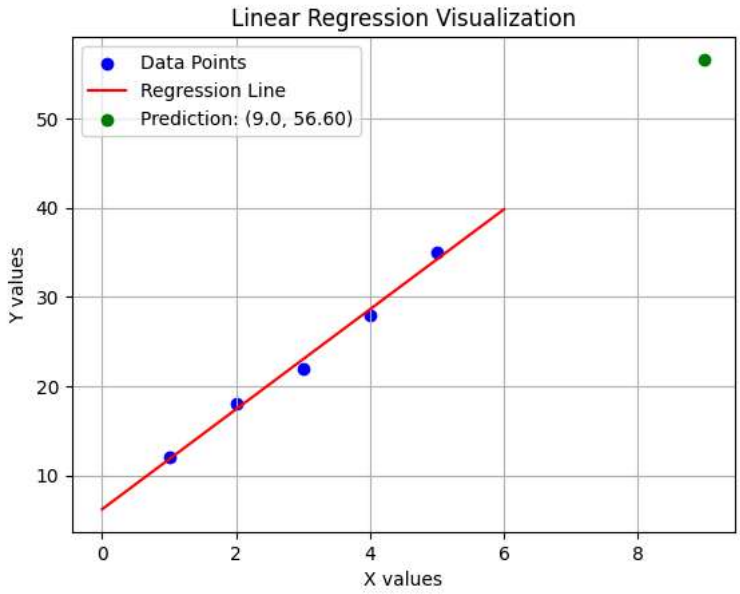
```
x = list(map(float, input("Enter X values separated by space: ").split()))
y = list(map(float, input("Enter Y values separated by space: ").split()))

n = len(x)

sum_x = sum(x)
sum_y = sum(y)
sum_x2 = sum(xi**2 for xi in x)
sum_xy = sum(x[i] * y[i] for i in range(n))

a1 = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x**2)
a0 = (sum_y - a1 * sum_x) / n

print("\nLinear Regression Equation:")
print(f"y = {a0:.2f} + {a1:.2f}x")

x_test = float(input("\nEnter a value of x to predict y: "))
y_pred = a0 + a1 * x_test
print(f"Predicted y: {y_pred:.2f}")



import matplotlib.pyplot as plt

x = list(map(float, input("Enter X values separated by space: ").split()))
y = list(map(float, input("Enter Y values separated by space: ").split()))

n = len(x)

sum_x = sum(x)
sum_y = sum(y)
sum_x2 = sum(xi**2 for xi in x)
sum_xy = sum(x[i] * y[i] for i in range(n))

a1 = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x**2)
a0 = (sum_y - a1 * sum_x) / n

print("\nLinear Regression Equation:")
print(f"y = {a0:.2f} + {a1:.2f}x")

x_test = float(input("\nEnter a value of x to predict y: "))
y_pred = a0 + a1 * x_test
print(f"Predicted y: {y_pred:.2f}")

plt.scatter(x, y, color='blue', label='Data Points')
01

x_line = [min(x) - 1, max(x) + 1]
y_line = [a0 + a1 * x_val for x_val in x_line]

plt.plot(x_line, y_line, color='red', label='Regression Line')
plt.scatter(x_test, y_pred, color='green', label=f'Prediction: ({x_test}, {y_pred:.2f})')

plt.xlabel('X values')
plt.ylabel('Y values')
plt.title('Linear Regression Visualization')
plt.legend()
plt.grid(True)
plt.show()
```

```
Enter X values separated by space: 1 2 3 4 5
Enter Y values separated by space: 12 18 22 28 35

Linear Regression Equation:
y = 6.20 + 5.60x

Enter a value of x to predict y: 9
Predicted y: 56.60
```



MATRIX FORM

```
import numpy as np


x = np.array(list(map(float, input("Enter X values separated by space: ").split())))
y = np.array(list(map(float, input("Enter Y values separated by space: ").split())))


X = np.c_[np.ones(len(x)), x]


beta = np.linalg.inv(X.T @ X) @ X.T @ y

print("\nLinear Regression Equation (Matrix Form):")
print(f"y = {beta[0]:.2f} + {beta[1]:.2f}x")


x_test = float(input("\nEnter a value of x to predict y: "))
y_pred = beta[0] + beta[1] * x_test
print(f"Predicted y: {y_pred:.2f}")
```

```
Enter X values separated by space: 1 2 3 4
Enter Y values separated by space: 1 3 4 8

Linear Regression Equation (Matrix Form):
y = -1.50 + 2.20x

Enter a value of x to predict y: 7
Predicted y: 13.90
```

```
import numpy as np
import matplotlib.pyplot as plt

x = np.array(list(map(float, input("Enter X values separated by space: ").split())))
y = np.array(list(map(float, input("Enter Y values separated by space: ").split())))

X = np.c_[np.ones(len(x)), x]
beta = np.linalg.inv(X.T @ X) @ X.T @ y
```

```
print("\nLinear Regression Equation (Matrix Form):")
print(f"y = {beta[0]:.2f} + {beta[1]:.2f}x")

x_test = float(input("\nEnter a value of x to predict y: "))
y_pred = beta[0] + beta[1] * x_test
print(f"Predicted y: {y_pred:.2f}")

plt.scatter(x, y, color='blue', label='Data Points')

x_line = np.linspace(min(x) - 1, max(x) + 1, 100)
y_line = beta[0] + beta[1] * x_line

plt.plot(x_line, y_line, color='red', label='Regression Line')
plt.scatter(x_test, y_pred, color='green', label=f'Prediction: ({x_test}, {y_pred:.2f})')

plt.xlabel('X values')
plt.ylabel('Y values')
plt.title('Linear Regression (Matrix Form)')
plt.legend()
plt.grid(True)
plt.show()
```

```
Enter X values separated by space: 1 2 3 4
Enter Y values separated by space: 1 3 4 8

Linear Regression Equation (Matrix Form):
y = -1.50 + 2.20x

Enter a value of x to predict y: 7
Predicted y: 13.90
```