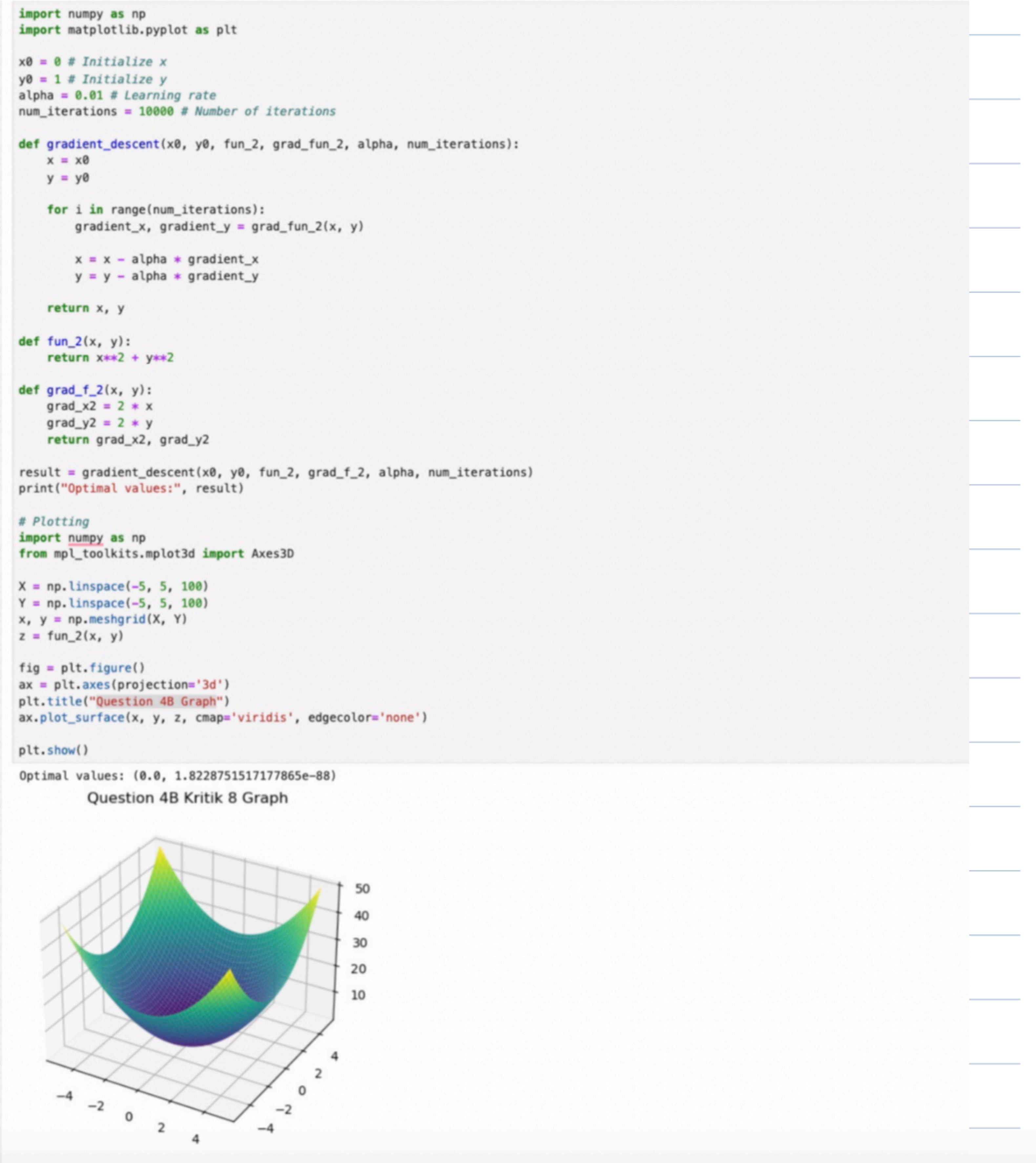

```
import numpy as np
import matplotlib.pyplot as plt

x0 = 0 # Initialize x
y0 = 1 # Initialize y
alpha = 0.01 # Learning rate
num_iterations = 10000 # Number of iterations

def gradient_descent(x0, y0, fun_2, grad_fun_2, alpha, num_iterations):
    x = x0
    y = y0

    for i in range(num_iterations):
        gradient_x, gradient_y = grad_fun_2(x, y)

        x = x - alpha * gradient_x
        y = y - alpha * gradient_y

    return x, y

def fun_2(x, y):
    return x**2 + y**2

def grad_f_2(x, y):
    grad_x2 = 2 * x
    grad_y2 = 2 * y
    return grad_x2, grad_y2

result = gradient_descent(x0, y0, fun_2, grad_f_2, alpha, num_iterations)
print("Optimal values:", result)

# Plotting
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

X = np.linspace(-5, 5, 100)
Y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(X, Y)
z = fun_2(x, y)

fig = plt.figure()
ax = plt.axes(projection='3d')
plt.title("Question 4B Graph")
ax.plot_surface(x, y, z, cmap='viridis', edgecolor='none')

plt.show()
```

```
Optimal values: (0.0, 1.8228751517177865e-88)
```

Question 4B Kritik 8 Graph