**import numpy as np**

**def function(x):**

**return x\*\*4 - 3 \* (x\*\*3) + 2**

**def derivative(x):**

**return 4 \* (x\*\*3) - 9 \* (x\*\*2)**

**def gradient\_descent(initial\_x, learning\_rate, precision, max\_iterations):**

**x = initial\_x**

**iteration = 0**

**while True:**

**gradient = derivative(x)**

**new\_x = x - learning\_rate \* gradient**

**if abs(new\_x - x) < precision:**

**print("Local minimum occurs at:", new\_x)**

**break**

**x = new\_x**

**iteration += 1**

**if iteration >= max\_iterations:**

**print("Exceeded the maximum number of iterations. No convergence.")**

**break**

**initial\_x = 6**

**learning\_rate = 0.01**

**precision = 0.0001**

**max\_iterations = 10000**

**gradient\_descent(initial\_x, learning\_rate, precision, max\_iterations)**