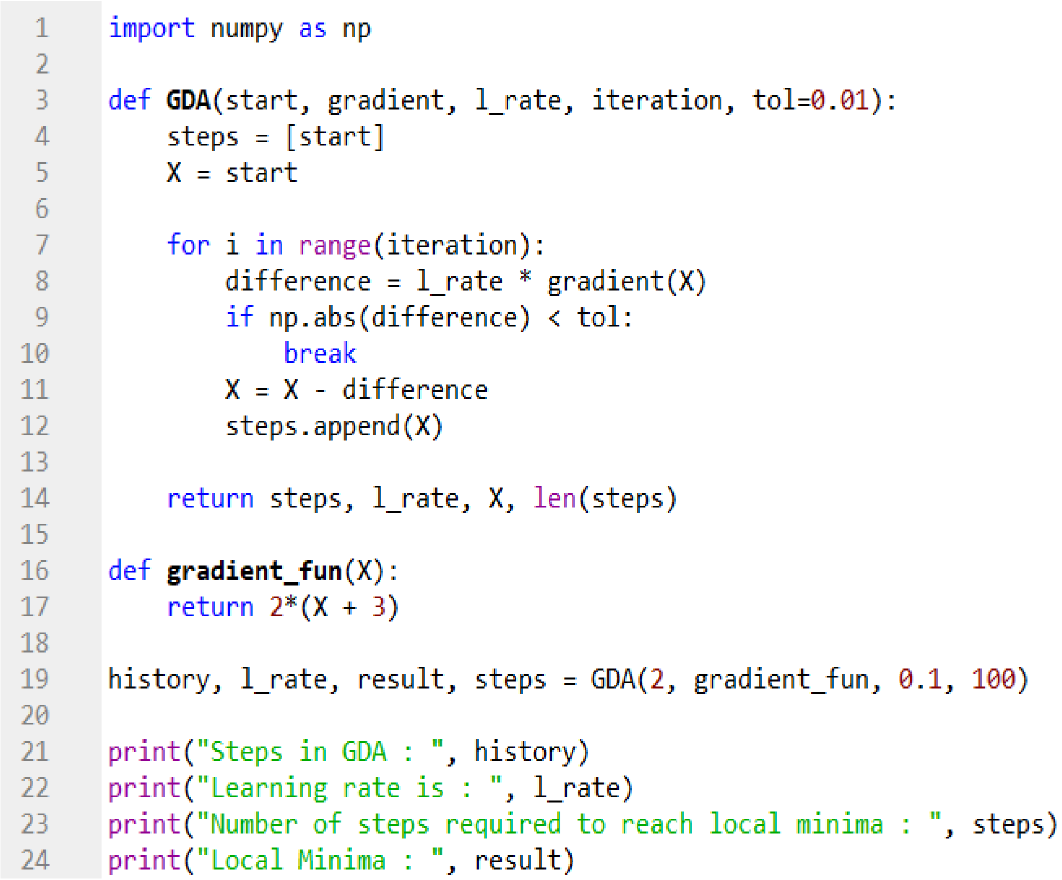
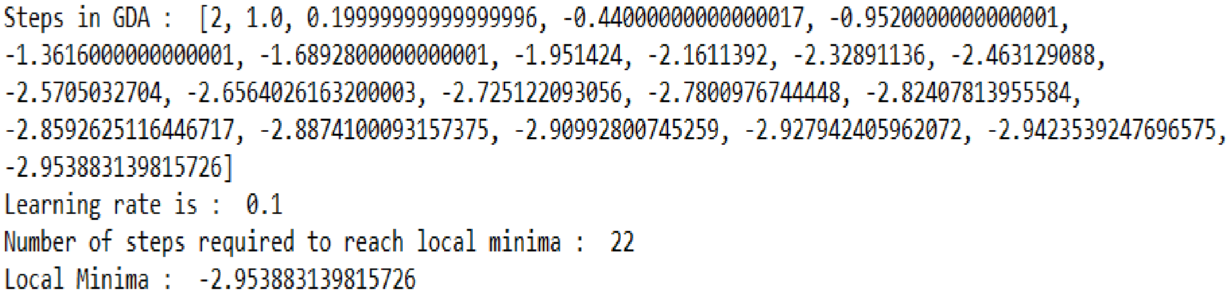


```python
import numpy as np

def GDA(start, gradient, l_rate, iteration, tol=0.01):
    steps = [start]
    X = start

    for i in range(iteration):
        difference = l_rate * gradient(X)
        if np.abs(difference) < tol:
            break
        X = X - difference
        steps.append(X)

    return steps, l_rate, X, len(steps)

def gradient_fun(X):
    return 2*(X + 3)

history, l_rate, result, steps = GDA(2, gradient_fun, 0.1, 100)

print("Steps in GDA : ", history)
print("Learning rate is : ", l_rate)
print("Number of steps required to reach local minima : ", steps)
print("Local Minima : ", result)
```

Output -



```
Steps in GDA :  [2, 1.0, 0.19999999999999996, -0.44000000000000017, -0.9520000000000001,
-1.3616000000000001, -1.6892800000000001, -1.951424, -2.1611392, -2.32891136, -2.463129088,
-2.5705032704, -2.6564026163200003, -2.725122093056, -2.7800976744448, -2.82407813955584,
-2.8592625116446717, -2.8874100093157375, -2.90992800745259, -2.927942405962072, -2.9423539247696575,
-2.953883139815726]
Learning rate is :  0.1
Number of steps required to reach local minima :  22
Local Minima :  -2.953883139815726
```