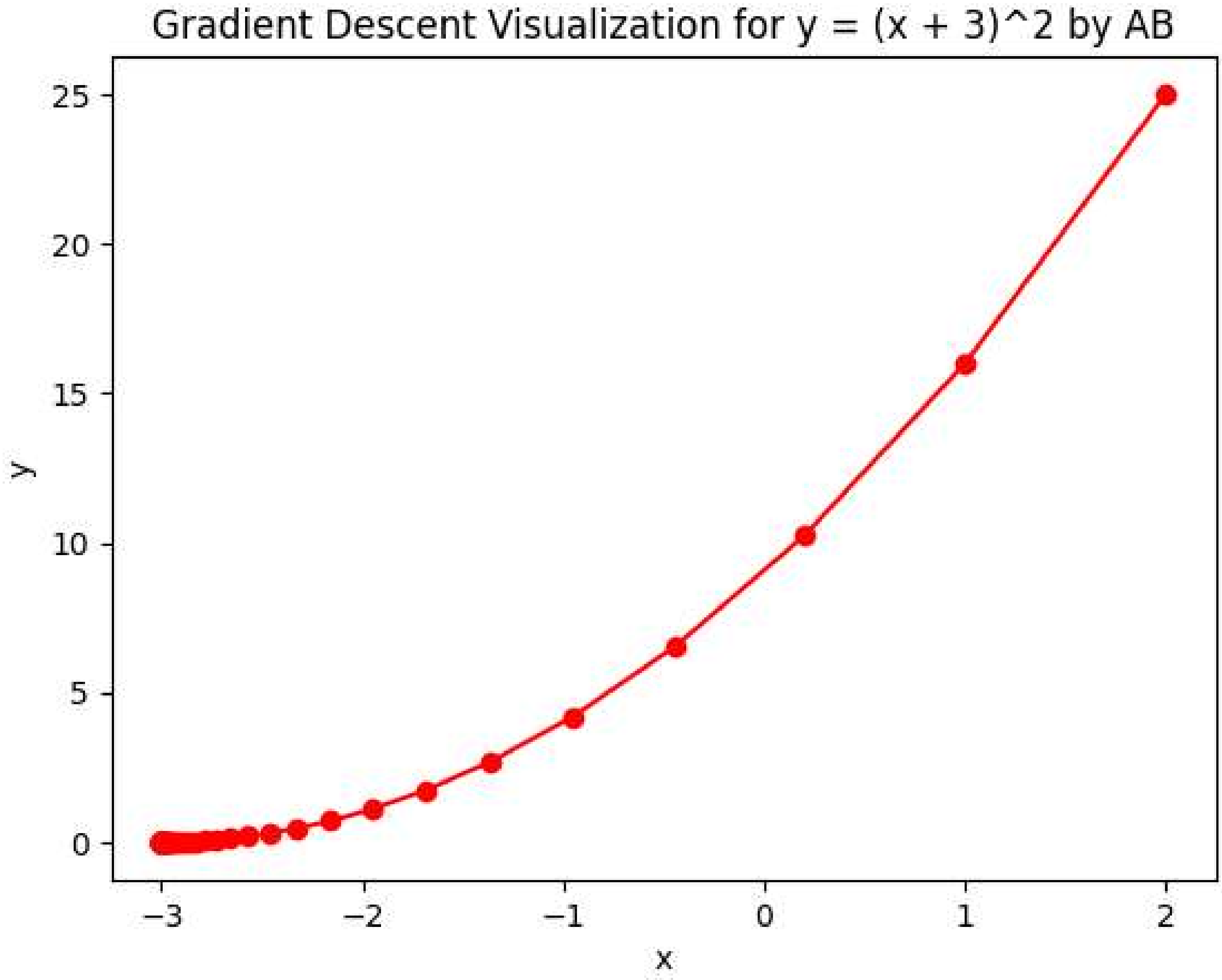
```
# Implement Gradient Descent Algorithm to find the local minima of a function.
# For example, find the local minima of the function y=(x+3)**2 starting from the poi

import matplotlib.pyplot as plt
def cost_function(x):
    # ithe given function yenar
    return (x + 3) ** 2
def gradient(x):
    # ithe derivate of given function yenar
    return 2 * (x + 3)
learning_rate = 0.1
initial_x = 2.0
num_iterations = 100
x_values = []
y_values = []
x = initial_x
for i in range(num_iterations):
    x_values.append(x)
    y_values.append(cost_function(x))
    gradient_value = gradient(x)
    x = x - learning_rate * gradient_value

    print(f'Iteration {i+1}: x = {x}, Cost = {cost_function(x)}')

print(f'Optimal x: {x}')
```

```
Iteration 1: x = 1.0, Cost = 16.0
Iteration 2: x = 0.19999999999999996, Cost = 10.240000000000002
Iteration 3: x = -0.44000000000000017, Cost = 6.553599999999998
Iteration 4: x = -0.9520000000000001, Cost = 4.194304
Iteration 5: x = -1.3616000000000001, Cost = 2.6843545599999996
Iteration 6: x = -1.6892800000000001, Cost = 1.7179869183999996
Iteration 7: x = -1.951424, Cost = 1.099511627776
Iteration 8: x = -2.1611392, Cost = 0.7036874417766399
Iteration 9: x = -2.32891136, Cost = 0.4503599627370493
Iteration 10: x = -2.463129088, Cost = 0.28823037615171165
Iteration 11: x = -2.5705032704, Cost = 0.1844674407370954
Iteration 12: x = -2.6564026163200003, Cost = 0.11805916207174093
Iteration 13: x = -2.725122093056, Cost = 0.07555786372591429
Iteration 14: x = -2.7800976744448, Cost = 0.04835703278458515
Iteration 15: x = -2.82407813955584, Cost = 0.030948500982134555
Iteration 16: x = -2.859262511644671 7, Cost = 0.019807040628566166
Iteration 17: x = -2.8874100093157375, Cost = 0.012676506002282305
Iteration 18: x = -2.90992800745259, Cost = 0.008112963841460692
Iteration 19: x = -2.927942405962072, Cost = 0.005192296858534868
Iteration 20: x = -2.9423539247696575, Cost = 0.0033230699894623056
Iteration 21: x = -2.953883139815726, Cost = 0.002126764793255884
Iteration 22: x = -2.9631065118525806, Cost = 0.0013611294676837786
Iteration 23: x = -2.9704852094820646, Cost = 0.0008711228593176078
Iteration 24: x = -2.9763881675856516, Cost = 0.0005575186299632732
Iteration 25: x = -2.981110534068521, Cost = 0.00035681192317650156
Iteration 26: x = -2.984888427254817, Cost = 0.00022835963083295564
Iteration 27: x = -2.9879107418038537, Cost = 0.00014615016373308945
Iteration 28: x = -2.990328593443083, Cost = 9.353610478917726e-05
Iteration 29: x = -2.9922628747544664, Cost = 5.986310706507345e-05
Iteration 30: x = -2.993810299803573, Cost = 3.83123885216492e-05
Iteration 31: x = -2.995048239842858, Cost = 2.451992865385725e-05
Iteration 32: x = -2.9960385918742864, Cost = 1.5692754338469342e-05
Iteration 33: x = -2.9968308734994293, Cost = 1.0043362776619255e-05
Iteration 34: x = -2.9974646987995435, Cost = 6.427752177036323e-06
Iteration 35: x = -2.997971759039635, Cost = 4.113761393302886e-06
Iteration 36: x = -2.998377407231708, Cost = 2.6328072917135587e-06
Iteration 37: x = -2.998701925785366, Cost = 1.6849966666971388e-06
Iteration 38: x = -2.998961540628293, Cost = 1.0783978666865378e-06
Iteration 39: x = -2.9991692325026342, Cost = 6.901746346793842e-07
Iteration 40: x = -2.9993353860021075, Cost = 4.417117661946878e-07
Iteration 41: x = -2.999468308801686, Cost = 2.826955303647891e-07
Iteration 42: x = -2.9995746470413485, Cost = 1.8092513943361614e-07
Iteration 43: x = -2.9996597176330786, Cost = 1.1579208923763523e-07
Iteration 44: x = -2.999727774106463, Cost = 7.410693711203819e-08
Iteration 45: x = -2.99978221928517, Cost = 4.7428439751781807e-08
Iteration 46: x = -2.9998257754281363, Cost = 3.035420144107846e-08
Iteration 47: x = -2.999860620342509, Cost = 1.9426688922339734e-08
Iteration 48: x = -2.999888496274007, Cost = 1.243308091029743e-08
Iteration 49: x = -2.9999107970192056, Cost = 7.9571717826062e-09
Iteration 50: x = -2.9999286376153647, Cost = 5.092589940842615e-09
Iteration 51: x = -2.9999429100922916, Cost = 3.259257562149415e-09
Iteration 52: x = -2.999954328073833, Cost = 2.0859248397837384e-09
Iteration 53: x = -2.9999634624590668, Cost = 1.3349918974486118e-09
Iteration 54: x = -2.9999707699672533, Cost = 8.543948143723039e-10
Iteration 55: x = -2.999976615973803, Cost = 5.468126811899669e-10
Iteration 56: x = -2.9999812927790424, Cost = 3.499601159582557e-10
Iteration 57: x = -2.9999850342232337, Cost = 2.2397447421860056e-10
```

```
plt.plot(x_values, y_values, 'ro-')
plt.title('Gradient Descent Visualization for y = (x + 3)^2 by AB')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```


Gradient Descent Visualization for y = (x + 3)^2 by AB
y
x