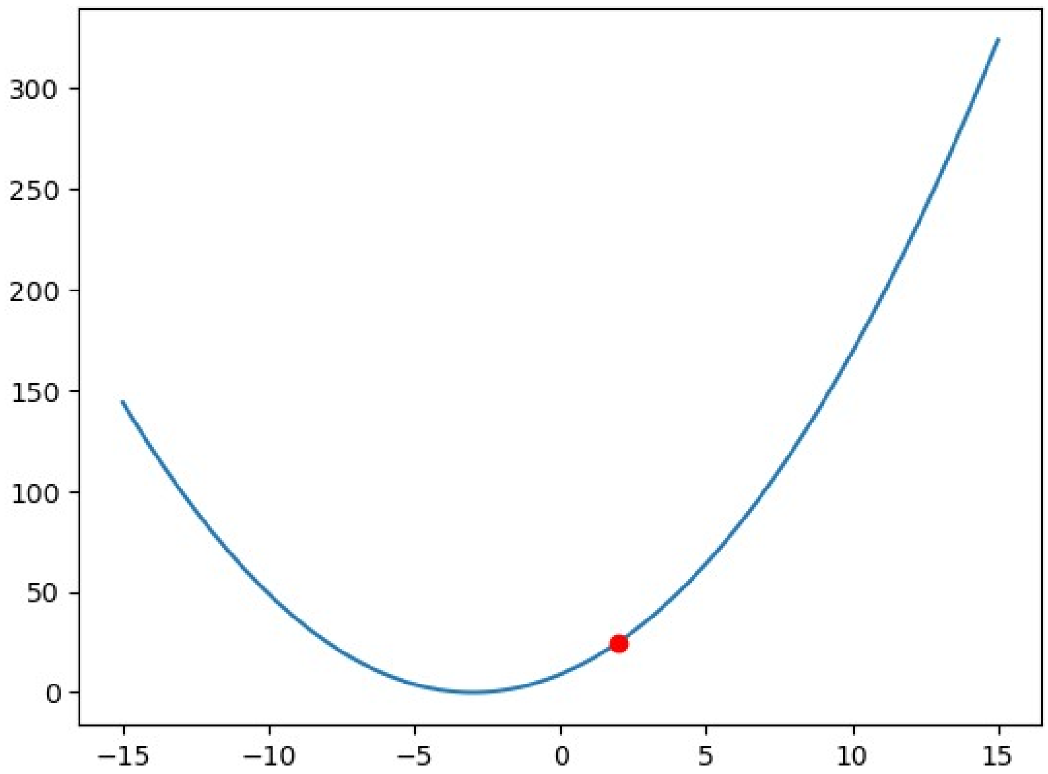
```python
# Title = 3.Implement Gradient Descent Algorithm to find the local
minima of a function. For example, find the local minima of the
function y=(x+3)² starting from the point x=2.
# Name = Tanmay Shrikrishna Badhe
# Div = B
# Roll No. 01

import numpy as np
import pandas as pd
import sympy as sym
import matplotlib as pyplot
from matplotlib import pyplot

def objective(x):
    return (x+3)**2

def derivative(x):
    return 2*(x+3)

def gradient(alpha,start,max_iter):
    x_list=list()
    x=start
    x_list.append(x)
    for i in range(max_iter):
        gradi=derivative(x)
        x=x-(alpha*gradi)
        x_list.append(x)
    return x_list
x=sym.symbols('x')
expr=(x+3)**2.0
grad=sym.Derivative(expr,x)
print("{}".format(grad.doit()))
grad.doit().subs(x,2)
```

```
2.0*(x + 3)**1.0
```

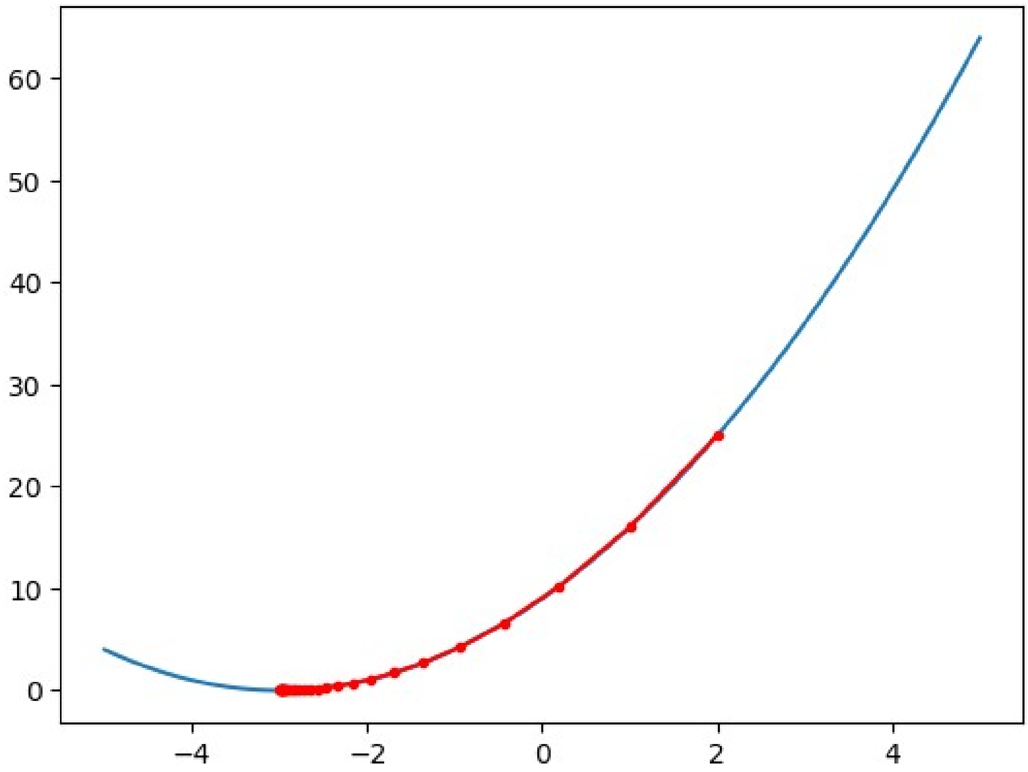
```
10.0000000000000
```

```python
alpha=0.1
start=2
max_iter=30
x=sym.symbols('x')
expr=(x+3)**2

x_cor=np.linspace(-15,15,100)
pyplot.plot(x_cor,objective(x_cor))
pyplot.plot(2,objective(2),'ro')
```

```
[<matplotlib.lines.Line2D at 0x253d8e2dff0>]
```



```
x=gradient(alpha,start,max_iter)
x_cor=np.linspace(-5,5,100)
pyplot.plot(x_cor,objective(x_cor))

x_arr=np.array(x)
pyplot.plot(x_arr,objective(x_arr),'.-',color='red')
pyplot.show()
```


60
50
40
30
20
10
0
−4
−2
0
2
4