```
# Problem Statement
# 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.
```

```
# Define the function y = (x + 3)²

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

# Define the derivative of the function
def gradient(x):
  return 2 *(x+3)

# Initialise starting point
x = 2

# Set Parameter
learning_rate = 0.1
num_iteration = 100

#Gradient Descent Algorithm
for i in range(num_iteration):
  #Calculate gradient at current point
  grad = gradient(x)

  #Update value x using gradient and learning rate
  x = x - learning_rate * grad

  if i % 10 == 0:
    print("Iteration:",i," x=",x,"y=",fun_minimise(x))

#Final value of x will be local minima
print("Local Minima: x= ", x,"y=",fun_minimise(x))
```

```
Iteration: 0  x= 1.0 y= 16.0
Iteration: 10  x= -2.5705032704 y= 0.1844674407370954
Iteration: 20  x= -2.953883139815726 y= 0.002126764793255884
Iteration: 30  x= -2.995048239842858 y= 2.451992865385725e-05
Iteration: 40  x= -2.999468308801686 y= 2.826955303647891e-07
Iteration: 50  x= -2.9999429100922916 y= 3.259257562149415e-09
Iteration: 60  x= -2.9999938700178364 y= 3.757668132666189e-11
Iteration: 70  x= -2.999999341798177 y= 4.3322963956744853e-13
Iteration: 80  x= -2.9999999293261177 y= 4.994797639633387e-15
Iteration: 90  x= -2.9999999924114498 y= 5.758609463330129e-17
Local Minima: x=  -2.999999998981482 y= 1.0373792396055266e-18
```