In [1]:

```
cur_x = 2 # The algorithm starts at x=2

rate = 0.01 # Learning rate

precision = 0.000001 #This tells us when to stop the algorithm

previous_step_size = 1

max_iters = 10000 # maximum number of iterations

iters = 0 #iteration counter

df = lambda x: 2*(x+3) # Gradient of our function y = (x+3)**2
```

In [2]:

```
while previous_step_size > precision and iters < max_iters:
    prev_x = cur_x                             #Store current x value in prev_
    cur_x = cur_x - rate * df(prev_x)          #Grad descent
    previous_step_size = abs(cur_x - prev_x)   #Change in x
    iters = iters+1                            #iteration count
    print("Iteration",iters,"\nX value is",cur_x) #Print iterations

print("The local minimum occurs at", cur_x)
```

```
Iteration 34
X value is -0.4843131601118455
Iteration 35
X value is -0.5346268969096086
Iteration 36
X value is -0.5839343589714164
Iteration 37
X value is -0.632255671791988
Iteration 38
X value is -0.6796105583561483
Iteration 39
X value is -0.7260183471890252
Iteration 40
X value is -0.7714979802452447
Iteration 41
X value is -0.8160680206403399
Iteration 42
X value is -0.8597466602275331
Iteration 43
X value is -0.9025517270229825
```

In [ ]: