

In [8]:

```
# O algoritmo inicia com o parâmetro x=3
cur_x = 6 # Valor aleatório

# Hiperparâmetro - Learning rate - Passo - Taxa de Aprendizagem
rate = 0.008 # 0.001 0.01 # 0.1 - 0.5 - 0.01 - ==>(Valor)

# Define quando parar o algoritmo
# Parar o loop quando a diferença entre os valores de x em 2 iterações consecutivas for menor
# ou quando o número de iterações exceder 10 mil
precision = 0.00000001 #0.0000000001 ==> f(cur_x) - f(prev_x)

# Inicializa o contador do passo anterior
previous_step_size = 1

# Número máximo de iterações - Hiperparametro
max_iters = 10000 # Número de Épocas (Epochs)

# Contador de iterações
iters = 0

# Gradiente da função  $(x + 5)^2$  = Derivada ->  $2 * (x + 5)$ 
df = lambda x: 2 * (x+5)
```

In [9]:

```
%%time
while previous_step_size > precision and iters < max_iters:

    # Armazena o valor corrente de x em prev_x
    prev_x = cur_x

    # Aplica o Gradient descent
    cur_x = cur_x - rate * df(prev_x)

    # Altera o valor de x
    previous_step_size = abs(cur_x - prev_x)

    # Incrementa o número de iterações
    iters = iters + 1

# Imprime as iterações
print("Iteration", iters, "\nValor de x igual a ", cur_x, ' | x - x_prev:', '{:.10f}'.f

print("\nO mínimo local da função ocorre em: ", cur_x)
```

```
Iteration 978
Valor de x igual a -4.999998449045879 | x - x_prev: 0.0000000252
Iteration 979
Valor de x igual a -4.999998473861146 | x - x_prev: 0.0000000248
Iteration 980
Valor de x igual a -4.999998498279368 | x - x_prev: 0.0000000244
Iteration 981
Valor de x igual a -4.999998522306898 | x - x_prev: 0.0000000240
Iteration 982
Valor de x igual a -4.999998545949987 | x - x_prev: 0.0000000236
Iteration 983
Valor de x igual a -4.999998569214787 | x - x_prev: 0.0000000233

Iteration 984
Valor de x igual a -4.99999859210735 | x - x_prev: 0.0000000229
Iteration 985
Valor de x igual a -4.999998614633633 | x - x_prev: 0.0000000225
Iteration 986
Valor de x igual a -4.999998636799495 | x - x_prev: 0.0000000222
Iteration 987
```