**Experiment 5**

**Program:-**

import numpy as np

def objective(x):

return -x[0] \*\* 2 + 5

def generate\_neighbors(x, step\_size=0.1):

return [np.array([x[0] + step\_size]), np.array([x[0] - step\_size])]

def hill\_climbing(objective, initial, n\_iterations=100, step\_size=0.1):

current = np.array([initial])

current\_eval = objective(current)

for i in range(n\_iterations):

neighbors = generate\_neighbors(current, step\_size)

neighbor\_evals = [objective(n) for n in neighbors]

best\_idx = np.argmax(neighbor\_evals)

if neighbor\_evals[best\_idx] > current\_eval:

current = neighbors[best\_idx]

current\_eval = neighbor\_evals[best\_idx]

print(

f"Step {i+1}: x = {current[0]:.4f}, f(x) = {current\_eval:.4f}")

else:

print("No better neighbors found. Algorithm converged.")

break

return current, current\_eval

initial\_guess = 2.0

solution, value = hill\_climbing(

objective, initial\_guess, n\_iterations=100, step\_size=0.1)

print(f"\nBest solution x = {solution[0]:.4f}, f(x) = {value:.4f}")

**Output:-**

****