import random

# Define your objective function. This function should return a score that the algorithm tries to maximize.

def objective\_function(solution):

# For demonstration purposes, let's use a simple example: maximizing the sum of a list of numbers.

return sum(solution)

# Define the hill climbing algorithm

def hill\_climbing(max\_iterations, problem\_size):

# Generate an initial random solution

current\_solution = [random.randint(0, 100) for \_ in range(problem\_size)]

current\_score = objective\_function(current\_solution)

for \_ in range(max\_iterations):

# Generate a neighbor solution by slightly perturbing the current solution

neighbor\_solution = current\_solution[:]

index\_to\_change = random.randint(0, problem\_size - 1)

neighbor\_solution[index\_to\_change] += random.randint(-10, 10)

# Calculate the score of the neighbor solution

neighbor\_score = objective\_function(neighbor\_solution)

# If the neighbor solution is better, move to it

if neighbor\_score > current\_score:

current\_solution = neighbor\_solution

current\_score = neighbor\_score

return current\_solution, current\_score

if \_\_name\_\_ == "\_\_main\_\_":

max\_iterations = 1000

problem\_size = 10

best\_solution, best\_score = hill\_climbing(max\_iterations, problem\_size)

print("Best Solution:", best\_solution)

print("Best Score:", best\_score)