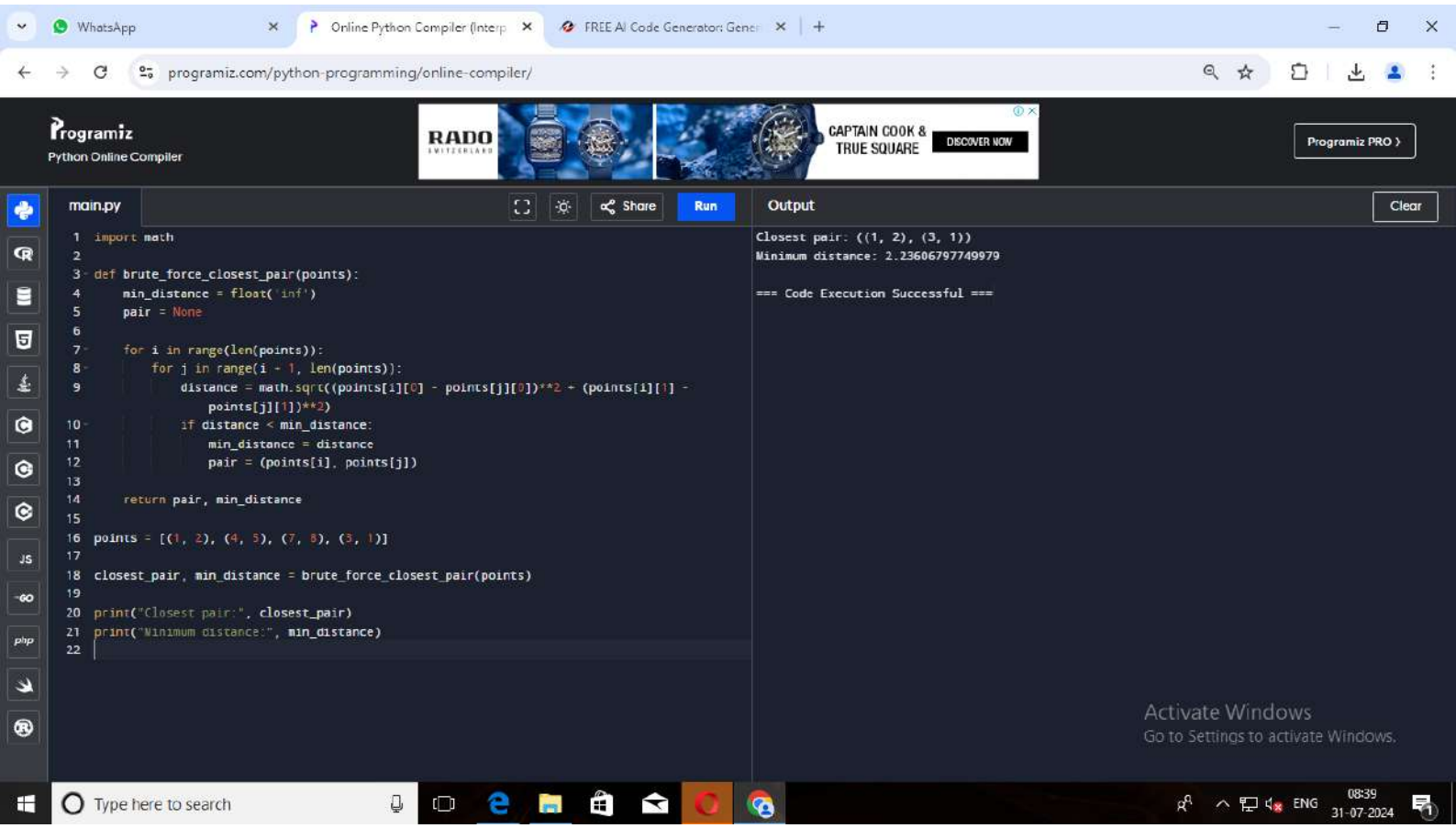
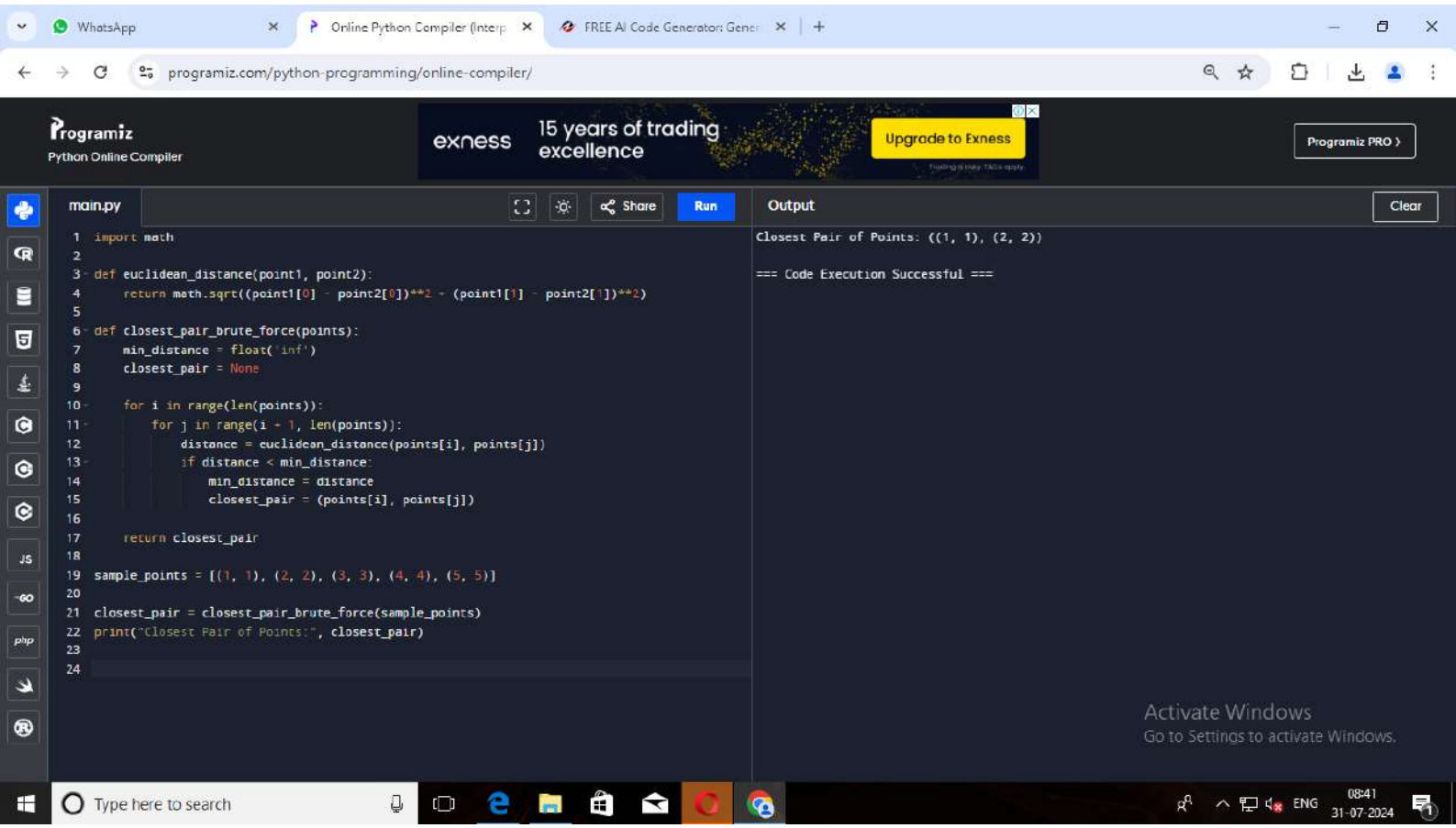
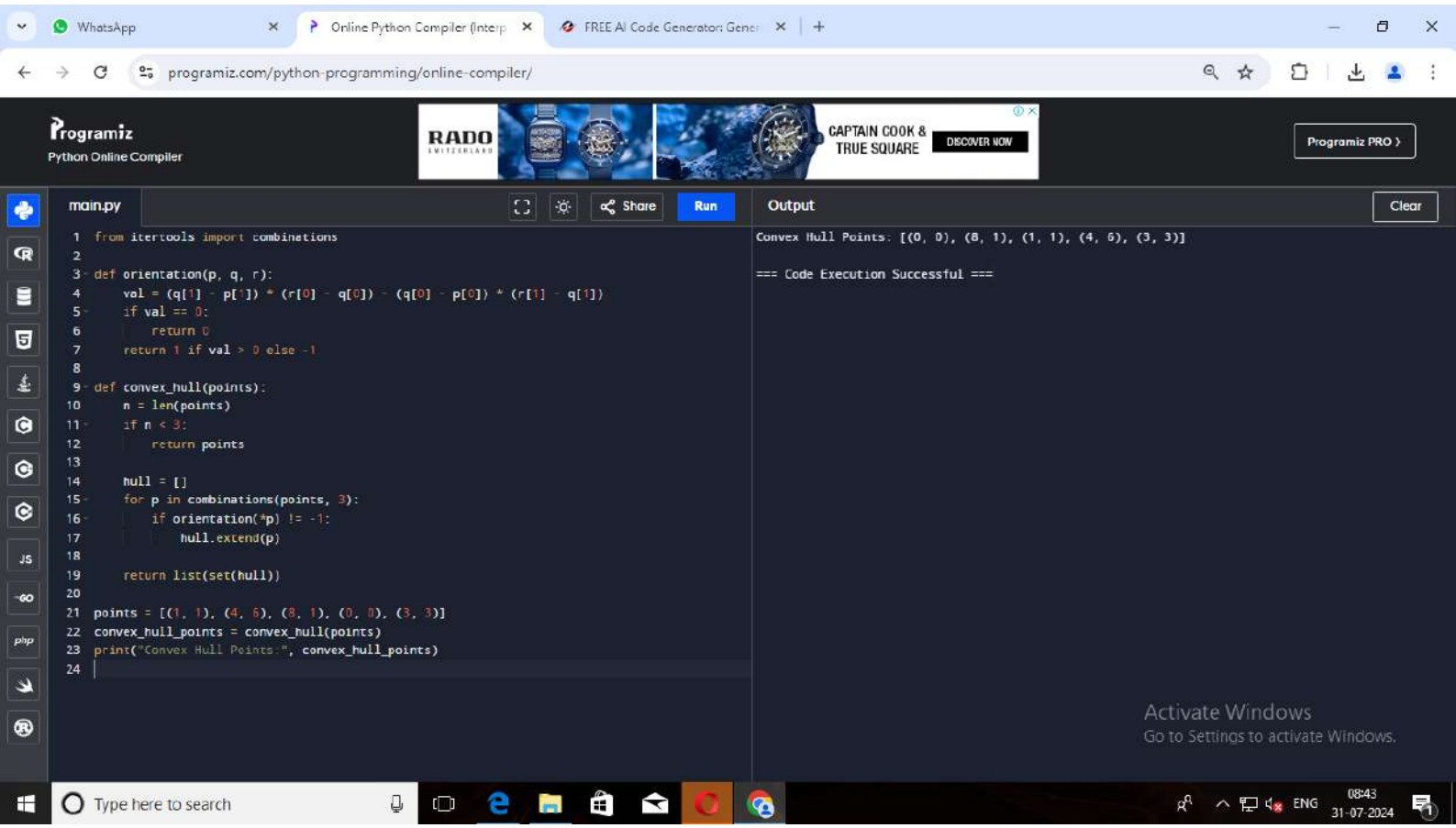
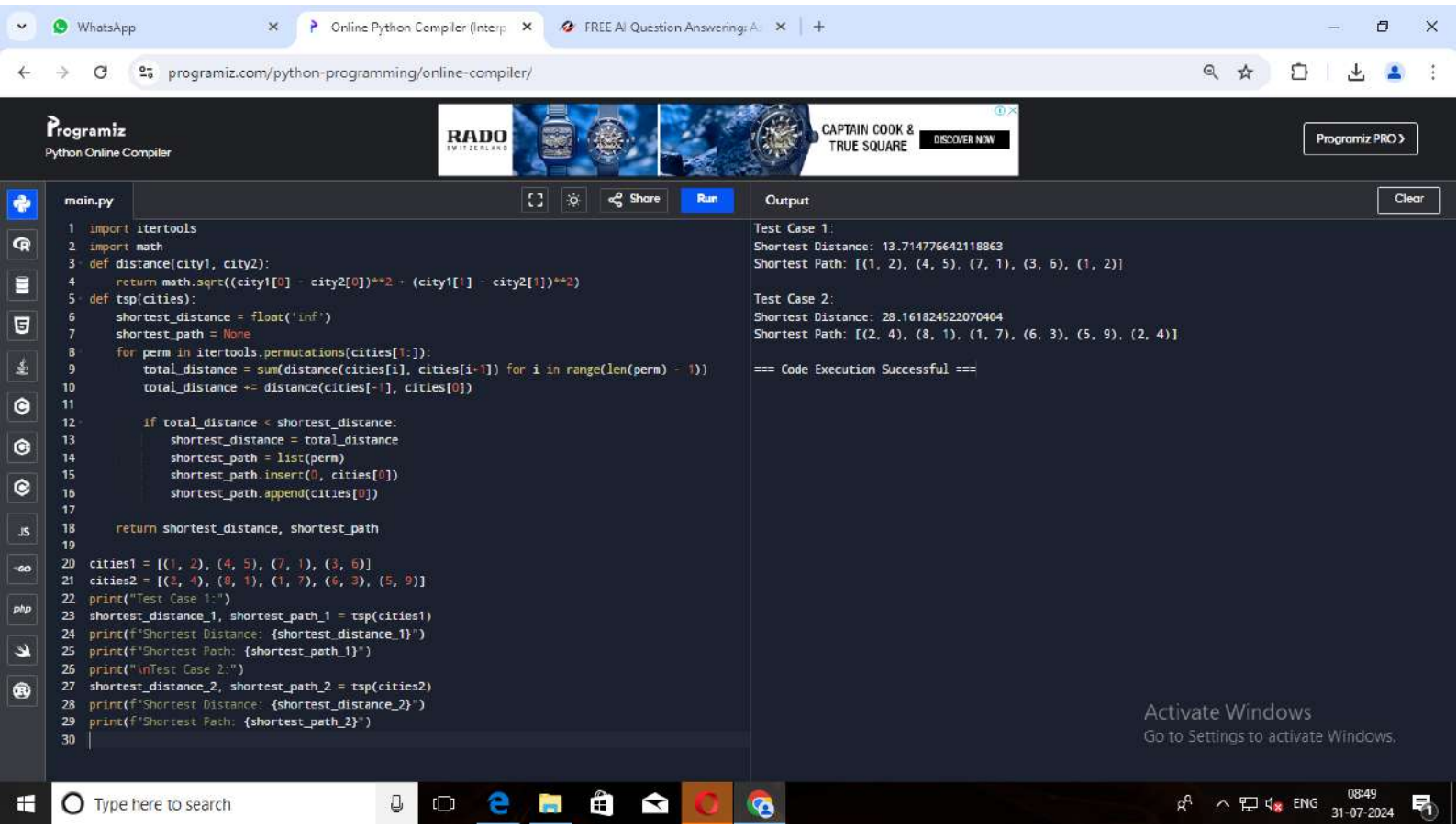
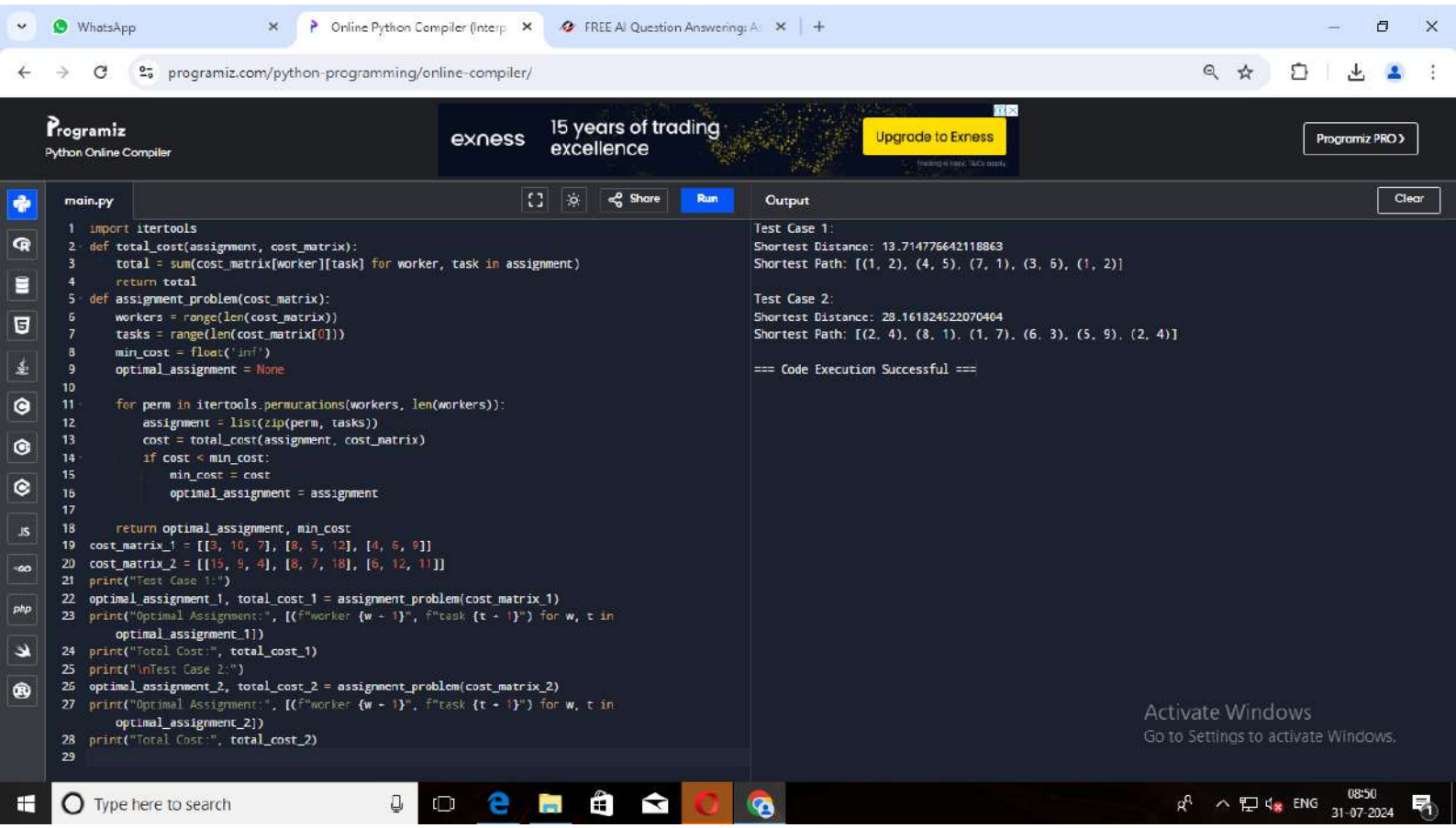
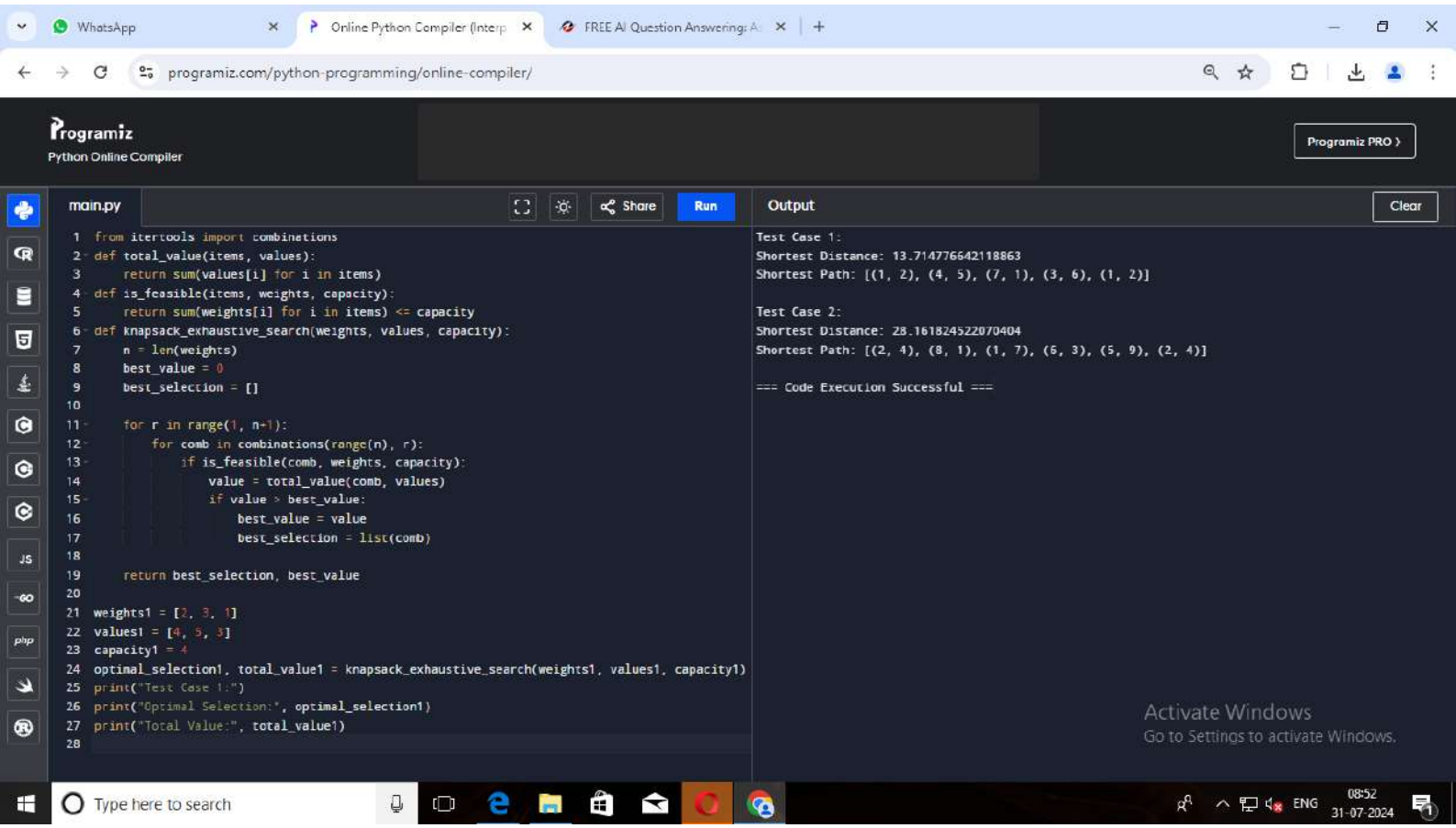

WhatsApp
Online Python Compiler (Interp
FREE AI Code Generator: Gener
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
RADO
SWITZERLAND
CAPTAIN COOK &
TRUE SQUARE
DISCOVER NOW
Programiz PRO ›
main.py
Share
Run
Output
Clear
import math
def brute_force_closest_pair(points):
min_distance = float('inf')
pair = None
for i in range(len(points)):
for j in range(i + 1, len(points)):
distance = math.sqrt((points[i][0] - points[j][0])**2 + (points[i][1] -
points[j][1])**2)
if distance < min_distance:
min_distance = distance
pair = (points[i], points[j])
return pair, min_distance
points = [(1, 2), (4, 5), (7, 8), (3, 1)]
closest_pair, min_distance = brute_force_closest_pair(points)
print("Closest pair:", closest_pair)
print("Minimum distance:", min_distance)
Closest pair: ((1, 2), (3, 1))
Minimum distance: 2.23606797749979
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:39
31-07-2024


WhatsApp
Online Python Compiler (Interp
FREE AI Code Generator: Gener
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
exness
15 years of trading excellence
Upgrade to Exness
Programiz PRO
main.py
Share
Run
Output
Clear
import math
def euclidean_distance(point1, point2):
return math.sqrt((point1[0] - point2[0])**2 + (point1[1] - point2[1])**2)
def closest_pair_brute_force(points):
min_distance = float('inf')
closest_pair = None
for i in range(len(points)):
for j in range(i + 1, len(points)):
distance = euclidean_distance(points[i], points[j])
if distance < min_distance:
min_distance = distance
closest_pair = (points[i], points[j])
return closest_pair
sample_points = [(1, 1), (2, 2), (3, 3), (4, 4), (5, 5)]
closest_pair = closest_pair_brute_force(sample_points)
print("Closest Pair of Points:", closest_pair)
Closest Pair of Points: ((1, 1), (2, 2))
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:41
31-07-2024


WhatsApp
Online Python Compiler (Interp
FREE AI Code Generator: Gener
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
RADO
SWITZERLAND
CAPTAIN COOK &
TRUE SQUARE
DISCOVER NOW
Programiz PRO ›
main.py
Share
Run
Output
Clear
from itertools import combinations
def orientation(p, q, r):
val = (q[1] - p[1]) * (r[0] - q[0]) - (q[0] - p[0]) * (r[1] - q[1])
if val == 0:
return 0
return 1 if val > 0 else -1
def convex_hull(points):
n = len(points)
if n < 3:
return points
hull = []
for p in combinations(points, 3):
if orientation(*p) != -1:
hull.extend(p)
return list(set(hull))
points = [(1, 1), (4, 6), (8, 1), (0, 0), (3, 3)]
convex_hull_points = convex_hull(points)
print("Convex Hull Points:", convex_hull_points)
Convex Hull Points: [(0, 0), (8, 1), (1, 1), (4, 6), (3, 3)]
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:43
31-07-2024


WhatsApp
Online Python Compiler (Interp
FREE AI Question Answering: A
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
RADO
SWITZERLAND
CAPTAIN COOK &
TRUE SQUARE
DISCOVER NOW
Programiz PRO
main.py
Share
Run
Output
Clear
import itertools
import math
def distance(city1, city2):
return math.sqrt((city1[0] - city2[0])**2 + (city1[1] - city2[1])**2)
def tsp(cities):
shortest_distance = float('inf')
shortest_path = None
for perm in itertools.permutations(cities[1:]):
total_distance = sum(distance(cities[i], cities[i-1]) for i in range(len(perm) - 1))
total_distance += distance(cities[-1], cities[0])
if total_distance < shortest_distance:
shortest_distance = total_distance
shortest_path = list(perm)
shortest_path.insert(0, cities[0])
shortest_path.append(cities[0])
return shortest_distance, shortest_path
cities1 = [(1, 2), (4, 5), (7, 1), (3, 6)]
cities2 = [(2, 4), (8, 1), (1, 7), (6, 3), (5, 9)]
print("Test Case 1:")
shortest_distance_1, shortest_path_1 = tsp(cities1)
print(f"Shortest Distance: {shortest_distance_1}")
print(f"Shortest Path: {shortest_path_1}")
print("\nTest Case 2:")
shortest_distance_2, shortest_path_2 = tsp(cities2)
print(f"Shortest Distance: {shortest_distance_2}")
print(f"Shortest Path: {shortest_path_2}")
Test Case 1:
Shortest Distance: 13.714776642118863
Shortest Path: [(1, 2), (4, 5), (7, 1), (3, 6), (1, 2)]
Test Case 2:
Shortest Distance: 28.161824522070404
Shortest Path: [(2, 4), (8, 1), (1, 7), (6, 3), (5, 9), (2, 4)]
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:49
31-07-2024


WhatsApp
Online Python Compiler (Interp
FREE AI Question Answering A
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
exness
15 years of trading excellence
Upgrade to Exness
Programiz PRO
main.py
Share
Run
Output
Clear
import itertools
def total_cost(assignment, cost_matrix):
total = sum(cost_matrix[worker][task] for worker, task in assignment)
return total
def assignment_problem(cost_matrix):
workers = range(len(cost_matrix))
tasks = range(len(cost_matrix[0]))
min_cost = float('inf')
optimal_assignment = None
for perm in itertools.permutations(workers, len(workers)):
assignment = list(zip(perm, tasks))
cost = total_cost(assignment, cost_matrix)
if cost < min_cost:
min_cost = cost
optimal_assignment = assignment
return optimal_assignment, min_cost
cost_matrix_1 = [[3, 10, 7], [8, 5, 12], [4, 6, 9]]
cost_matrix_2 = [[15, 9, 4], [8, 7, 18], [6, 12, 11]]
print("Test Case 1:")
optimal_assignment_1, total_cost_1 = assignment_problem(cost_matrix_1)
print("Optimal Assignment:", [(f"worker {w + 1}", f"task {t + 1}") for w, t in optimal_assignment_1])
print("Total Cost:", total_cost_1)
print("\nTest Case 2:")
optimal_assignment_2, total_cost_2 = assignment_problem(cost_matrix_2)
print("Optimal Assignment:", [(f"worker {w + 1}", f"task {t + 1}") for w, t in optimal_assignment_2])
print("Total Cost:", total_cost_2)
Test Case 1:
Shortest Distance: 13.714776642118863
Shortest Path: [(1, 2), (4, 5), (7, 1), (3, 6), (1, 2)]
Test Case 2:
Shortest Distance: 28.161824522070404
Shortest Path: [(2, 4), (8, 1), (1, 7), (6, 3), (5, 9), (2, 4)]
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:50
31-07-2024


WhatsApp
Online Python Compiler (Interp
FREE AI Question Answering A
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Programiz PRO ›
main.py
Share
Run
Output
Clear
from itertools import combinations
def total_value(items, values):
return sum(values[i] for i in items)
def is_feasible(items, weights, capacity):
return sum(weights[i] for i in items) <= capacity
def knapsack_exhaustive_search(weights, values, capacity):
n = len(weights)
best_value = 0
best_selection = []
for r in range(1, n+1):
for comb in combinations(range(n), r):
if is_feasible(comb, weights, capacity):
value = total_value(comb, values)
if value > best_value:
best_value = value
best_selection = list(comb)
return best_selection, best_value
weights1 = [2, 3, 1]
values1 = [4, 5, 3]
capacity1 = 4
optimal_selection1, total_value1 = knapsack_exhaustive_search(weights1, values1, capacity1)
print("Test Case 1:")
print("Optimal Selection:", optimal_selection1)
print("Total Value:", total_value1)
Test Case 1:
Shortest Distance: 13.714776642118863
Shortest Path: [(1, 2), (4, 5), (7, 1), (3, 6), (1, 2)]
Test Case 2:
Shortest Distance: 28.161824522070404
Shortest Path: [(2, 4), (8, 1), (1, 7), (6, 3), (5, 9), (2, 4)]
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:52
31-07-2024