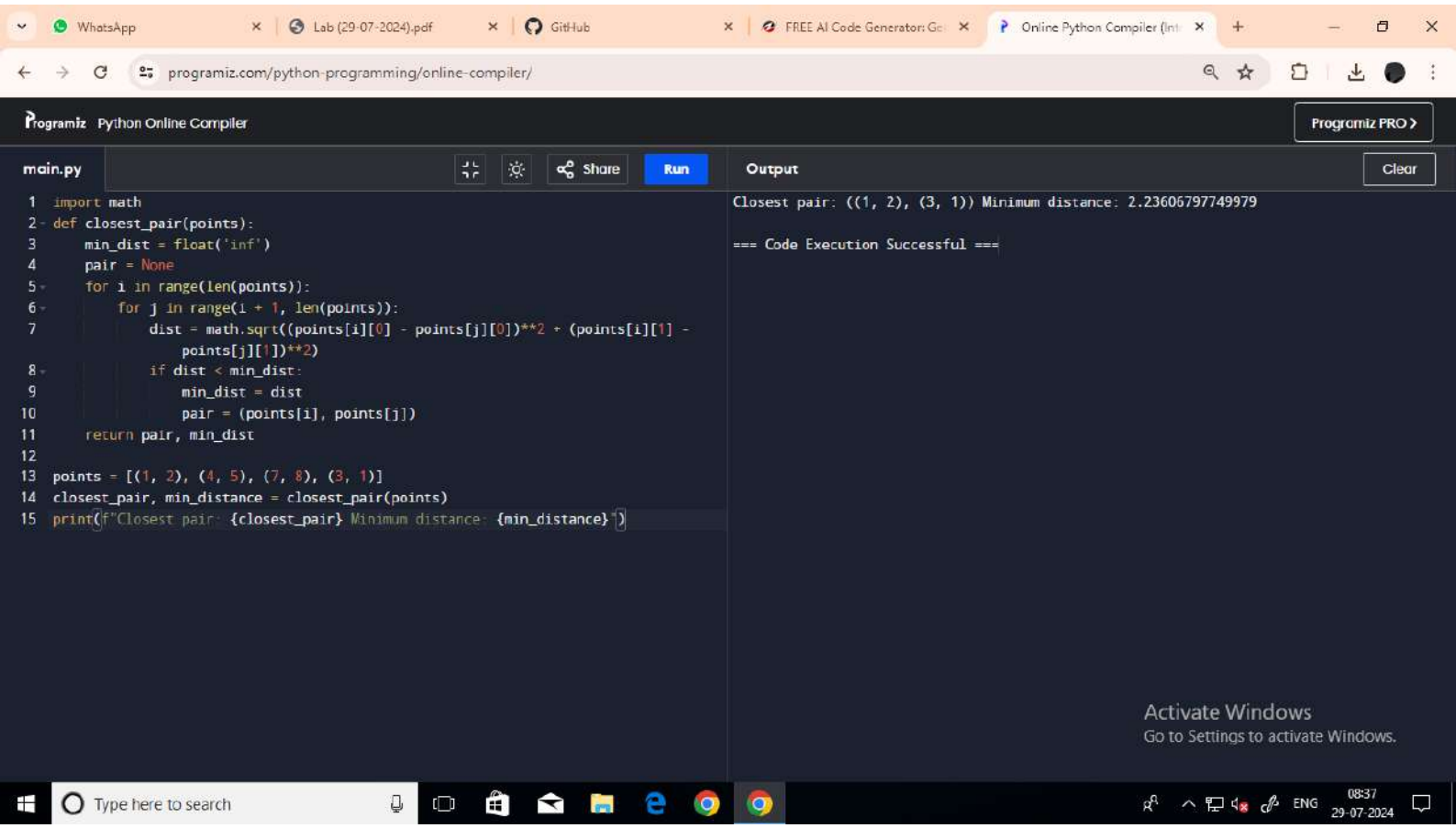
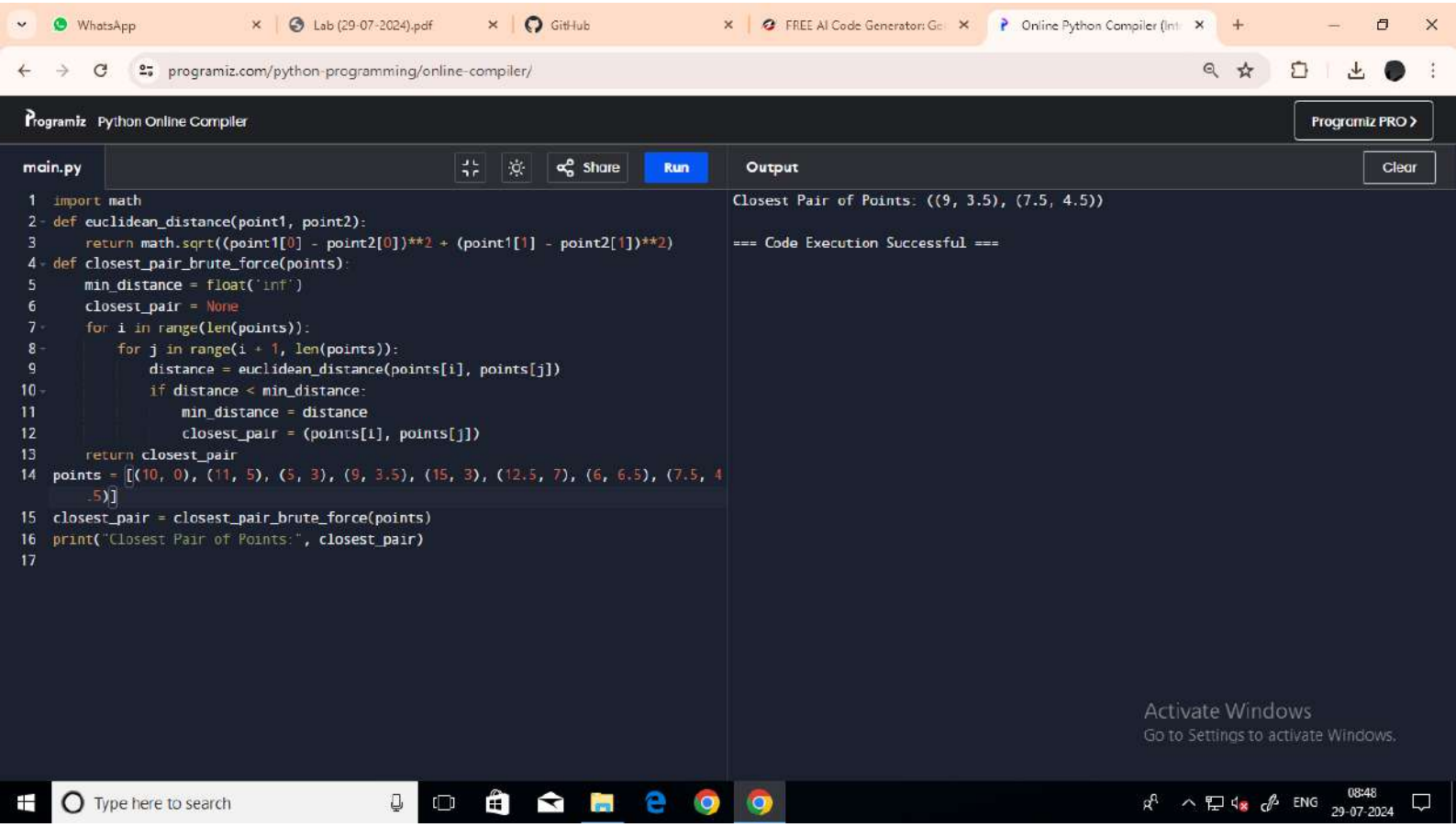
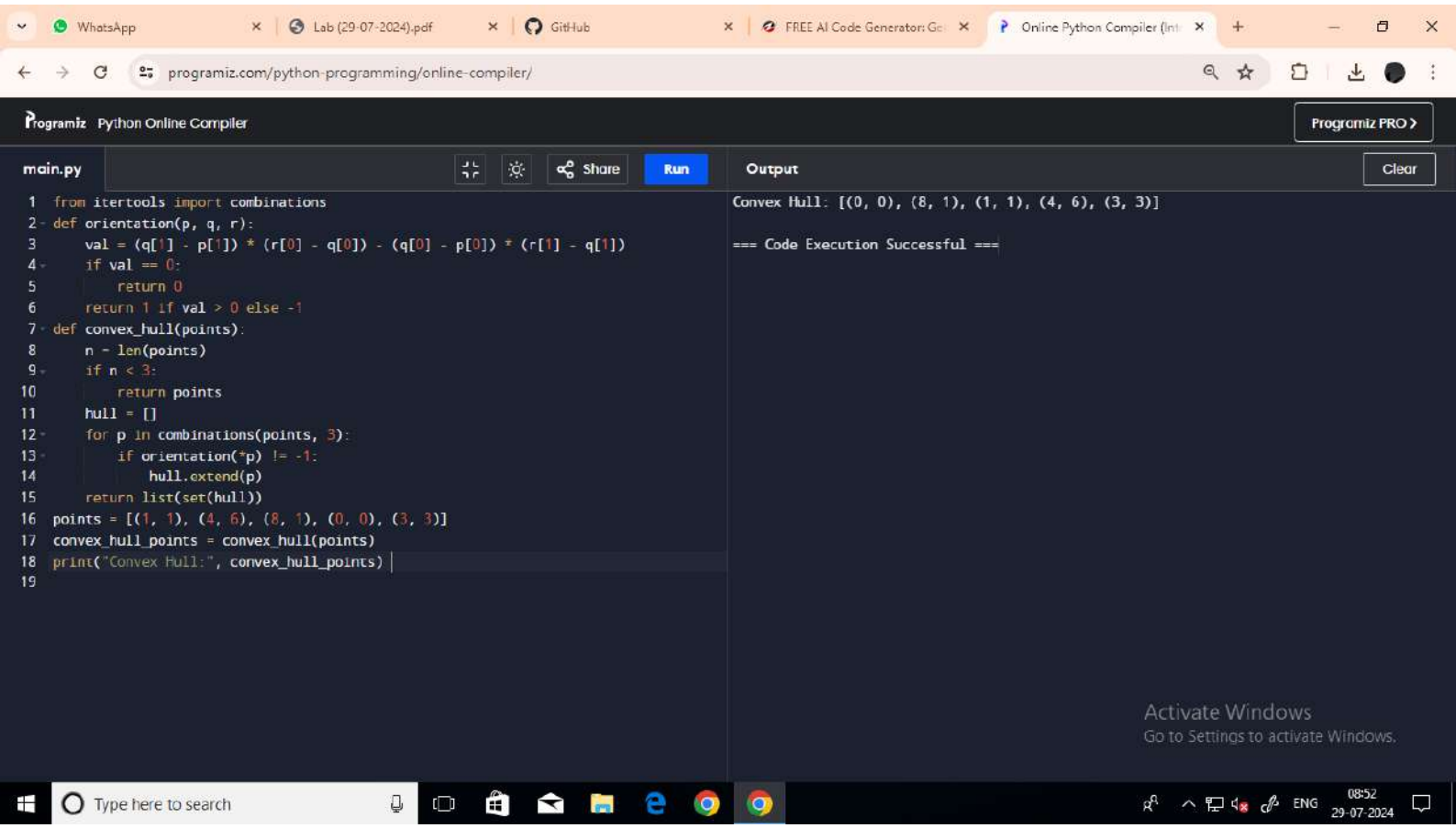
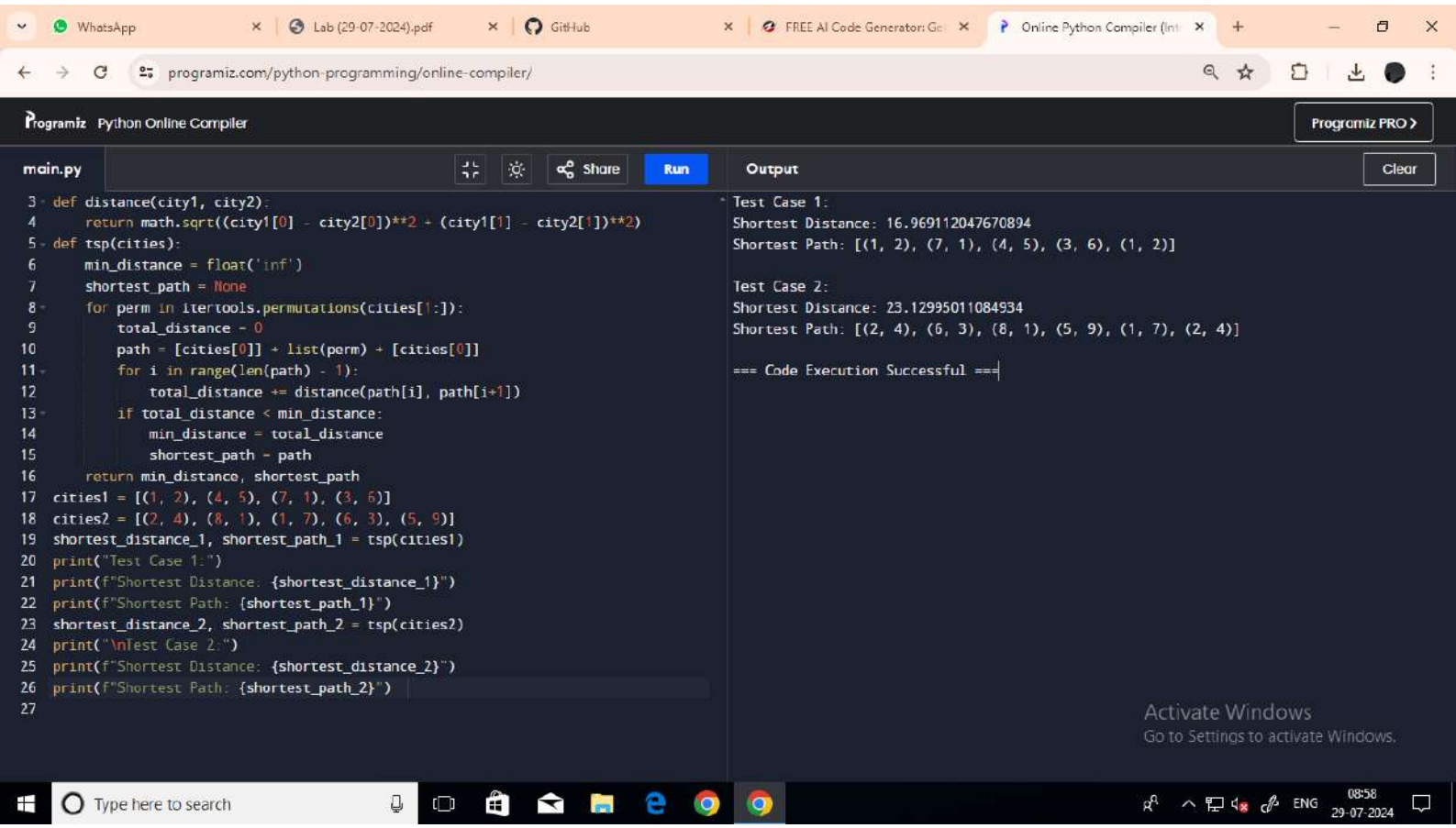
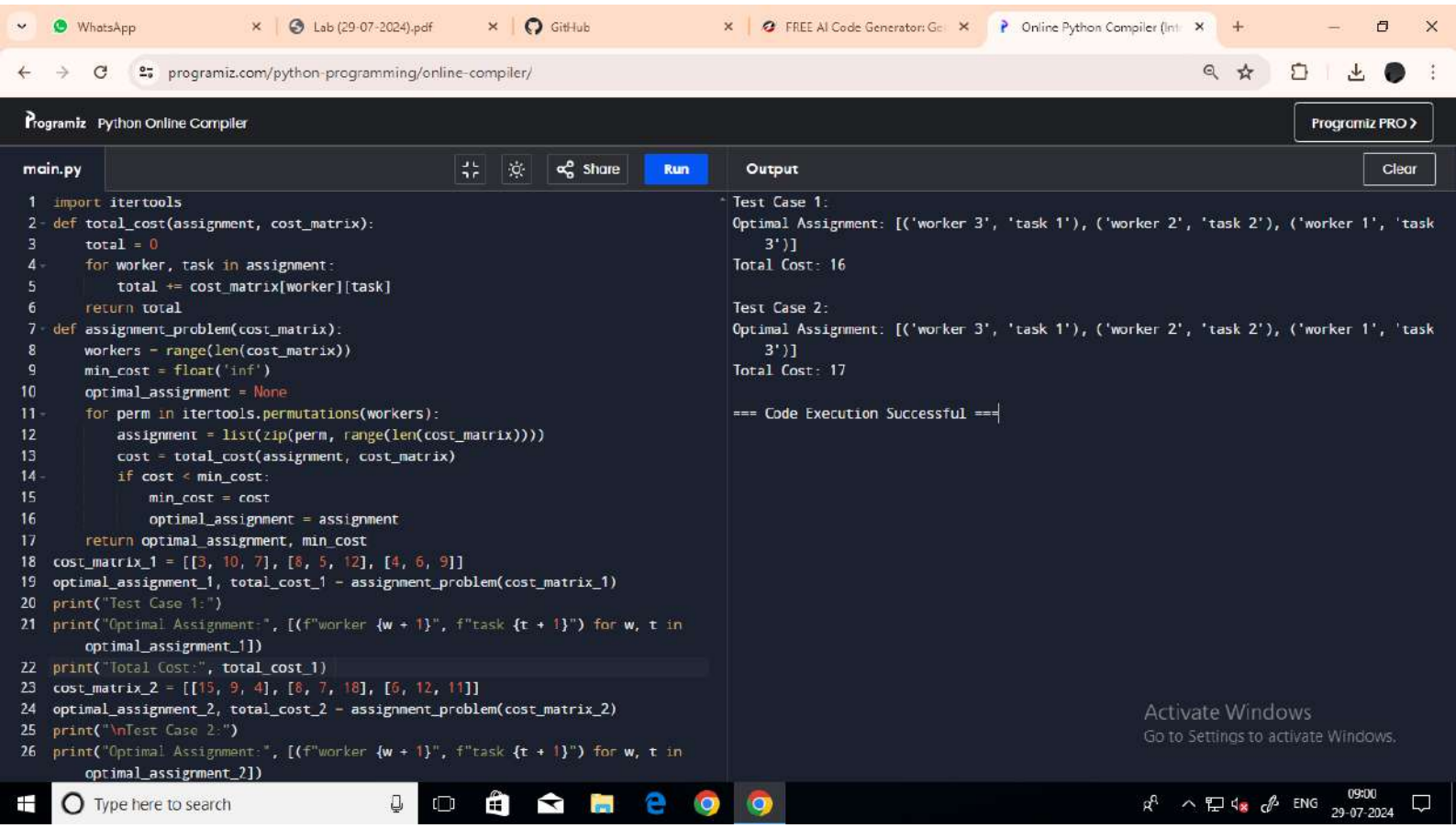
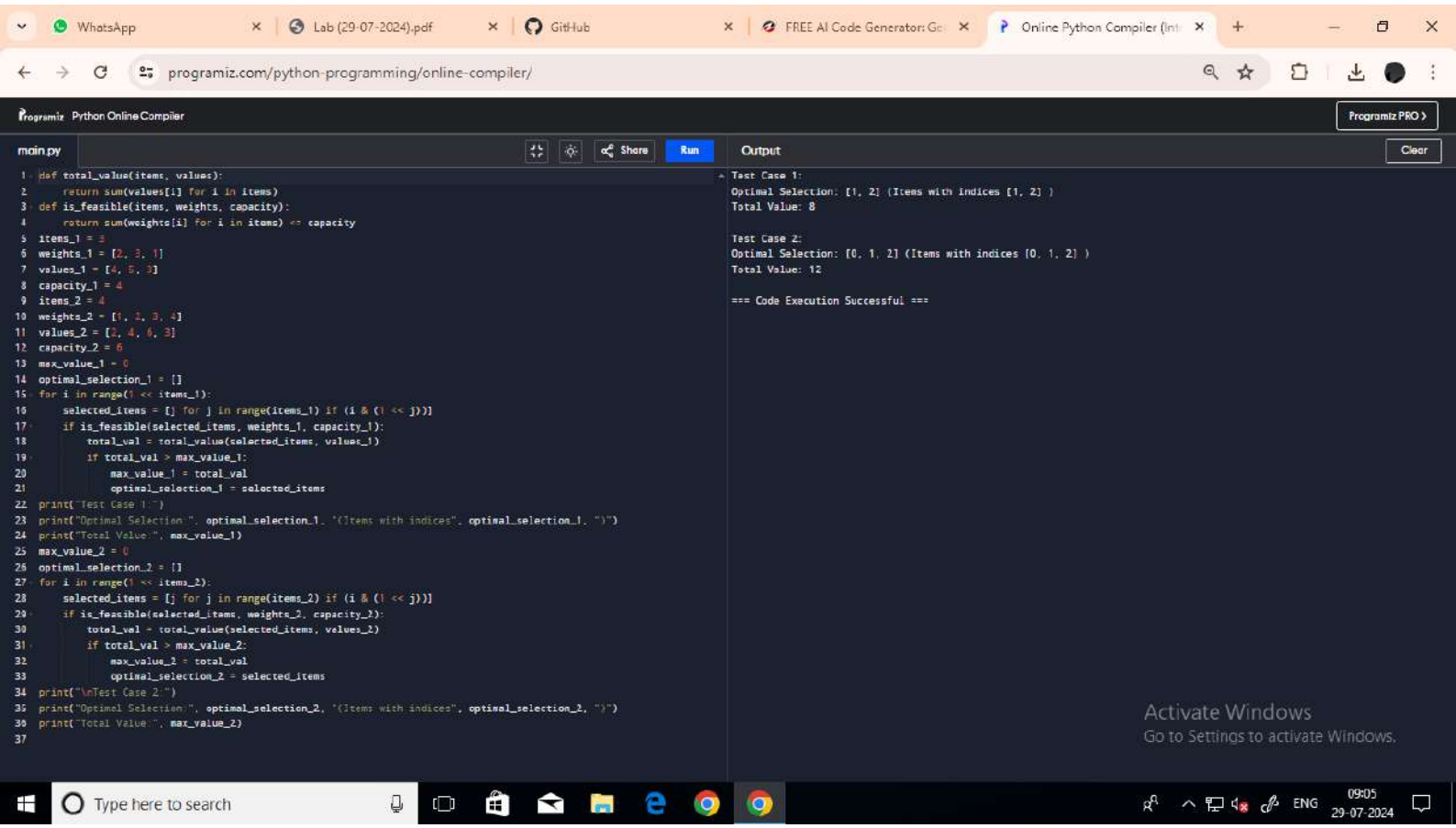

WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
import math
def closest_pair(points):
min_dist = float('inf')
pair = None
for i in range(len(points)):
for j in range(i + 1, len(points)):
dist = math.sqrt((points[i][0] - points[j][0])**2 + (points[i][1] -
points[j][1])**2)
if dist < min_dist:
min_dist = dist
pair = (points[i], points[j])
return pair, min_dist
points = [(1, 2), (4, 5), (7, 8), (3, 1)]
closest_pair, min_distance = closest_pair(points)
print(f"Closest pair: {closest_pair} 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
08:37
29-07-2024


WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
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
points = [(10, 0), (11, 5), (5, 3), (9, 3.5), (15, 3), (12.5, 7), (6, 6.5), (7.5, 4
.5)]
closest_pair = closest_pair_brute_force(points)
print("Closest Pair of Points:", closest_pair)
Closest Pair of Points: ((9, 3.5), (7.5, 4.5))
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:48
29-07-2024


WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
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:", convex_hull_points)
Convex Hull: [(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
08:52
29-07-2024


WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
def distance(city1, city2):
return math.sqrt((city1[0] - city2[0])**2 + (city1[1] - city2[1])**2)
def tsp(cities):
min_distance = float('inf')
shortest_path = None
for perm in itertools.permutations(cities[1:]):
total_distance = 0
path = [cities[0]] + list(perm) + [cities[0]]
for i in range(len(path) - 1):
total_distance += distance(path[i], path[i+1])
if total_distance < min_distance:
min_distance = total_distance
shortest_path = path
return min_distance, shortest_path
cities1 = [(1, 2), (4, 5), (7, 1), (3, 6)]
cities2 = [(2, 4), (8, 1), (1, 7), (6, 3), (5, 9)]
shortest_distance_1, shortest_path_1 = tsp(cities1)
print("Test Case 1:")
print(f"Shortest Distance: {shortest_distance_1}")
print(f"Shortest Path: {shortest_path_1}")
shortest_distance_2, shortest_path_2 = tsp(cities2)
print("\nTest Case 2:")
print(f"Shortest Distance: {shortest_distance_2}")
print(f"Shortest Path: {shortest_path_2}")
Test Case 1:
Shortest Distance: 16.969112047670894
Shortest Path: [(1, 2), (7, 1), (4, 5), (3, 6), (1, 2)]
Test Case 2:
Shortest Distance: 23.12995011084934
Shortest Path: [(2, 4), (6, 3), (8, 1), (5, 9), (1, 7), (2, 4)]
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
08:58
29-07-2024


WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
import itertools
def total_cost(assignment, cost_matrix):
total = 0
for worker, task in assignment:
total += cost_matrix[worker][task]
return total
def assignment_problem(cost_matrix):
workers = range(len(cost_matrix))
min_cost = float('inf')
optimal_assignment = None
for perm in itertools.permutations(workers):
assignment = list(zip(perm, range(len(cost_matrix))))
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]]
optimal_assignment_1, total_cost_1 = assignment_problem(cost_matrix_1)
print("Test Case 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)
cost_matrix_2 = [[15, 9, 4], [8, 7, 18], [6, 12, 11]]
optimal_assignment_2, total_cost_2 = assignment_problem(cost_matrix_2)
print("\nTest Case 2:")
print("Optimal Assignment:", [(f"worker {w + 1}", f"task {t + 1}") for w, t in optimal_assignment_2])
Test Case 1:
Optimal Assignment: [('worker 3', 'task 1'), ('worker 2', 'task 2'), ('worker 1', 'task 3')]
Total Cost: 16
Test Case 2:
Optimal Assignment: [('worker 3', 'task 1'), ('worker 2', 'task 2'), ('worker 1', 'task 3')]
Total Cost: 17
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
09:00
29-07-2024


WhatsApp
Lab (29-07-2024).pdf
GitHub
FREE AI Code Generator: Ge
Online Python Compiler (Int
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
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
items_1 = 3
weights_1 = [2, 3, 1]
values_1 = [4, 5, 3]
capacity_1 = 4
items_2 = 4
weights_2 = [1, 2, 3, 4]
values_2 = [2, 4, 6, 3]
capacity_2 = 6
max_value_1 = 0
optimal_selection_1 = []
for i in range(1 << items_1):
selected_items = [j for j in range(items_1) if (i & (1 << j))]
if is_feasible(selected_items, weights_1, capacity_1):
total_val = total_value(selected_items, values_1)
if total_val > max_value_1:
max_value_1 = total_val
optimal_selection_1 = selected_items
print("Test Case 1:")
print("Optimal Selection:", optimal_selection_1, "(Items with indices", optimal_selection_1, ")")
print("Total Value:", max_value_1)
max_value_2 = 0
optimal_selection_2 = []
for i in range(1 << items_2):
selected_items = [j for j in range(items_2) if (i & (1 << j))]
if is_feasible(selected_items, weights_2, capacity_2):
total_val = total_value(selected_items, values_2)
if total_val > max_value_2:
max_value_2 = total_val
optimal_selection_2 = selected_items
print("\nTest Case 2:")
print("Optimal Selection:", optimal_selection_2, "(Items with indices", optimal_selection_2, ")")
print("Total Value:", max_value_2)
Test Case 1:
Optimal Selection: [1, 2] (Items with indices [1, 2])
Total Value: 8
Test Case 2:
Optimal Selection: [0, 1, 2] (Items with indices [0, 1, 2])
Total Value: 12
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
09:05
29-07-2024