**2.13** You are given a cost matrix where each element cost[i][j] represents the cost of assigning worker i to task j. Develop a program that utilizes **exhaustive search** to solve the **assignment problem**. The program should Define a function total\_cost(assignment, cost\_matrix) that takes an assignment (list representing worker-task pairings) and the cost matrix as input. It iterates through the assignment and calculates the total cost by summing the corresponding costs from the cost matrix Implement a function assignment\_problem(cost\_matrix) that takes the cost matrix as input and performs the following Generate all possible permutations of worker indices (excluding repetitions).

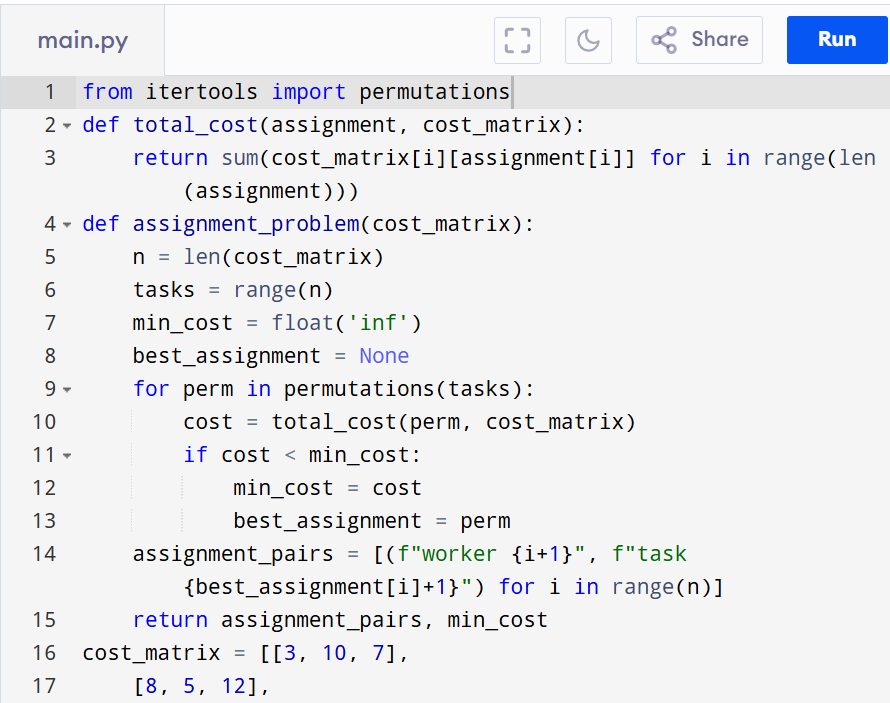
**AIM**

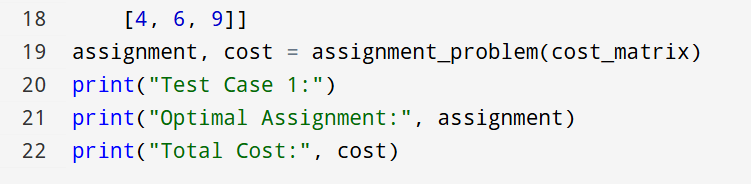
To determine the optimal assignment of workers to tasks that results in the minimum total cost, using the exhaustive search method.

**ALGORITHM**

1. **Start**
2. Define total\_cost(assignment, cost\_matrix) to Iterate over each worker index and its assigned task.
3. Sum up the cost from cost\_matrix[worker][task].
4. Define assignment\_problem(cost\_matrix) to Generate all permutations of task indices using itertools.permutations.
5. For each permutation: Calculate total cost using total\_cost.
6. Keep track of the lowest cost and corresponding assignment.
7. Return the optimal assignment and minimum total cost.
8. · **End**

**PROGRAM**





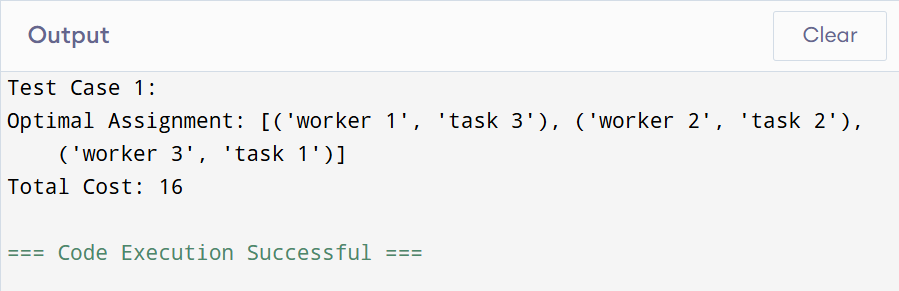
Input:

[[3, 10, 7],

[8, 5, 12],

[4, 6, 9]]

Output:



**RESULT:**

Thus, Solving the Assignment Problem using Exhaustive Search is successfully executed and the output is verified.

**PERFORMANCE ANALYSIS:**

· **Time Complexity:** O(n!)

· **Space Complexity:** O(n)