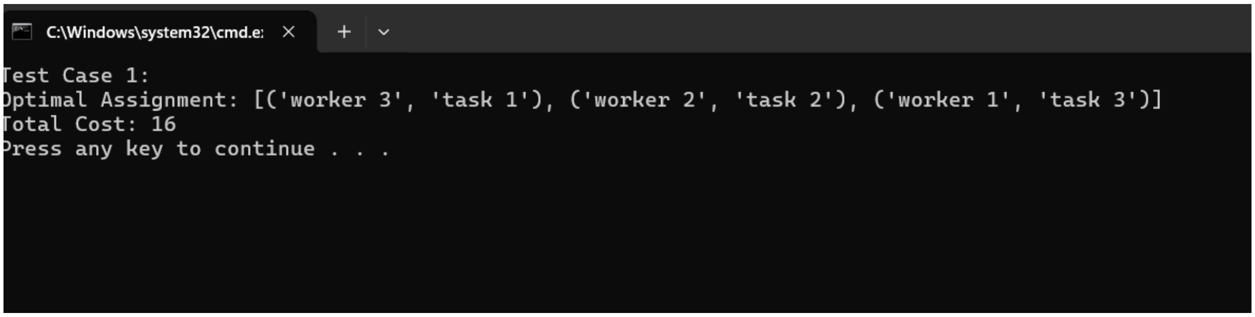
143)Assignment problem

CODE:

```
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):
        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_matrix1 = [[3, 10, 7], [8, 5, 12], [4, 6, 9]]
optimal_assignment1, total_cost1 = assignment_problem(cost_matrix1)
print("Test Case 1:")
print("Optimal Assignment:", [(f"worker {w+1}", f"task {t+1}") for w, t in
optimal_assignment1])
print("Total Cost:", total_cost1)
```

OUTPUT:



```
Test Case 1:
Optimal Assignment: [('worker 3', 'task 1'), ('worker 2', 'task 2'), ('worker 1', 'task 3')]
Total Cost: 16
Press any key to continue . . .
```

TIME COMPLEXITY : O(n*n!)