**Experiment No: 9**

**Name:Shubham Kaspate**

**Roll No:20141241**

**Batch:I3**

**Implement any scheme to find the optimal solution for the traveling salesperson problem- Code-**

// CPP program to implement traveling salesman problem #include <bits/stdc++.h>

using namespace std; #define V 4

int travllingSalesmanProblem(int graph[][V], int s)

{

vector<int> vertex; for (int i = 0; i < V; i++)

if (i != s)

vertex.push\_back(i); int min\_path = INT\_MAX;

do {

int current\_pathweight = 0; int k = s;

for (int i = 0; i < vertex.size(); i++) { current\_pathweight += graph[k][vertex[i]]; k = vertex[i];

}

current\_pathweight += graph[k][s];

min\_path = min(min\_path, current\_pathweight);

} while (

next\_permutation(vertex.begin(), vertex.end())); return min\_path;

}

int main()

{

int graph[][V] = { { 0, 8, 10, 20 },

{ 10, 0, 25, 25 },

{ 15, 35, 0, 30 },

{ 20, 25, 30, 0 } };

int s = 0;

cout << travllingSalesmanProblem(graph, s) << endl; return 0;

}

Output-

