```
#include <bits/stdc++.h>
#define inf INT_MAX
using namespace std;

void line(int l)
{
    for (int i = 0; i < l; i++)
        cout << "_";
    cout << endl;
}

void print_cost(vector<vector<int>> cost, int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (cost[i][j] == inf)
                cout << "X\t";
            else
                cout << cost[i][j] << "\t";
        }
        cout << endl;
    }
}

void fillinf(vector<vector<int>> &cost, vector<int> trav, int n, int s, int d)
{
    cost[d][s] = inf;
    for (int t : trav)
        cost[d][t] = inf;
```

```
    for (int i = 0; i < n; i++)
    {
        cost[i][d] = inf;
        cost[s][i] = inf;
    }
}

int reduce(vector<vector<int>> &cost, int n)
{
    int r = 0;
    for (int i = 0; i < n; i++)
    {
        int row_min = cost[i][0];
        for (int j = 1; j < n; j++)
            row_min = min(cost[i][j], row_min);

        if (row_min != inf)
        {
            r += row_min;
            for (int j = 0; j < n; j++)
                if (cost[i][j] != inf)
                    cost[i][j] -= row_min;
        }
    }
    for (int j = 0; j < n; j++)
    {
        int col_min = cost[0][j];
        for (int i = 1; i < n; i++)
            col_min = min(cost[i][j], col_min);

        if (col_min != inf)
```

```
        {
            r += col_min;
            for (int i = 0; i < n; i++)
                if (cost[i][j] != inf)
                    cost[i][j] -= col_min;
        }
    }
    return r;
}

void tsp(vector<vector<int>> source, int n, int start)
{

    line(50);
    cout << "\n\n\nTraveling Salesman Problem\nTotal locations = " << n << "\tStarting location = " <<
start << "\n\n\n";
    line(50);
    int s = start;
    vector<int> trav;
    int tcost = reduce(source, n);

    while (trav.size() != n)
    {
        int d = s;
        int min_cost = inf;
        vector<vector<int>> new_source = source;
        cout << "Current location = " << s << "\tCost = " << tcost << endl;
        line(50);
        print_cost(source, n);
        line(50);
        cout << "Possible next locations\n";
```

```
line(50);
for (int i = 0; i < n; i++)
{
   if (i == s || count(trav.begin(), trav.end(), i))
      continue;
   vector<vector<int>> dest = source;
   fillinf(dest, trav, n, s, i);
   int r = reduce(dest, n);
   int c = tcost + source[s][i] + r;
   print_cost(dest, n);
   cout << "\nLocation = " << i << "\tCost"
       << " = " << tcost << " + " << source[s][i] << " + " << r << " = " << c << endl;
   line(50);
   if (min_cost > c)
   {
      d = i;
      min_cost = c;
      new_source = dest;
   }
}

if (s == d)
   d = start;
cout << "Traveled to location = " << d << "\tCost"
    << " = " << tcost << endl;

line(50);
trav.push_back(s);
s = d;
if (min_cost != inf)
   tcost = min_cost;
```

```
        source = new_source;
    }

    cout << "Solution\nCost\t" << tcost << "\nPath\t";
    for (int t : trav)
        cout << t << " => ";
    cout << start << endl;
}

int main()
{
    int n, start = 0;
    cout << "Enter number of locations >>>";
    cin >> n;
    vector<vector<int>> source(n, vector<int>(n, inf));
    cout << "Enter cost adjacency matrix\n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            if (i != j)
                cin >> source[i][j];

    while (start >= 0 && start < n)
    {
        line(50);
        cout << "Enter starting location >>>";
        cin >> start;
        tsp(source, n, start);
    }
}
```

```
// vector<vector<int>> c1 = {
//     {inf, 20, 30, 10, 11},
//     {15, inf, 16, 4, 2},
//     {3, 5, inf, 2, 4},
//     {19, 6, 18, inf, 3},
//     {16, 4, 7, 16, inf}};



//     20  30  10  11
// 15      16  4   2
// 3   5       2   4
// 19  6   18      3
// 16  4   7   16

// 0 3 1 4 2
```

OUTPUT ::

```
Enter number of locations >>>4
Enter cost adjacency matrix
10 15 20
10 35 25
15 35 30
20 25 30
```

```
Enter starting location >>>0
```

```
Traveling Salesman Problem
Total locations = 4     Starting location = 0
```

```
Current location = 0    Cost = 70
```

X 0 0 0

0 X 20 5

0 20 X 5

0 5 5 X

Possible next locations

X X X X

X X 10 0

0 X X 5

0 X 0 X

Location = 1 Cost = 70 + 0 + 10 = 80

X X X X

0 X X 5

X 10 X 0

0 0 X X

Location = 2 Cost = 70 + 0 + 10 = 80

X X X X

0 X 20 X

0 20 X X

X 0 0 X

Location = 3 Cost = 70 + 0 + 5 = 75

Traveled to location = 3 Cost = 70

Current location = 3 Cost = 75

X X X X

0 X 20 X

0 20 X X

X 0 0 X

Possible next locations

X X X X

X X 0 X

0 X X X

X X X X

Location = 1 Cost = 75 + 0 + 20 = 95

X X X X

0 X X X

X 0 X X

X X X X

Location = 2 Cost = 75 + 0 + 20 = 95

Traveled to location = 1 Cost = 75

Current location = 1 Cost = 95

X X X X

X X 0 X

0 X X X

X X X X

Possible next locations

X X X X

X X X X

X X X X

X X X X

Location = 2 Cost = 95 + 0 + 0 = 95

Traveled to location = 2 Cost = 95

Current location = 2 Cost = 95

X X X X

X X X X

X X X X

X X X X

Possible next locations

Traveled to location = 0 Cost = 95

Solution

Cost 95

Path 0 => 3 => 1 => 2 => 0
