

```

#include<iostream>
#include<set>
#include<iomanip>
#include<vector>

#define INF 999
using namespace std;

class TSP{
    int n;
    vector< vector<int> > c;
public:
    TSP(int n){
        this->n = n;
        c.resize(n, vector<int> (n,INF) );
        for(int i=0;i<n;i++) c[i][i] = 0;
    }

    void takeInput(){
        int ne;
        cout << "Enter number of edges: ";
        cin >> ne;

        while(ne--){
            int u,v,w;
            cin >> u >> v >> w;

            u--; v--; c[u][v] = w;
        }
    }

    void run(){
        set<int> s;
        for(int i=0;i<n;i++) s.insert(i);

        int minCost = findMin(0,s);
        cout << "Minimum cost: " << minCost << endl;
    }

    int findMin(int st, set<int>& s){

        if(s.empty()) return c[st][0];

        int ans = INF;
        for(auto x: s){
            set<int> ss = s; ss.erase(x);
            int tmp = c[st][x] + findMin(x,ss);

            ans = min(ans,tmp);
        }

        return ans;
    }

    void printCost(){
        cout << endl << " - - - Printing cost matrix - - -" << endl;
        for(int i=0;i<n;i++){

```

```
        for(int j=0;j<n;j++) {
            if(c[i][j] == INF)
                cout << setw(6) << "INF";
            else
                cout << setw(6) << c[i][j];
        }
        cout << endl;
    }
    cout << endl;
};
```

```
int main(){

    int n;
    cout << "Number of nodes: ";
    cin >> n;

    TSP tsp(n);
    tsp.takeInput();
    tsp.printCost();
    tsp.run();

    return 0;
}
```