OBJECTIVE

**Perform Travelling Salesman Problem….**

METHODOLOGY

**The goal of the problem is, given a list of cities and their locations, to find the shortest possible path that passes through each city exactly once and ends at the starting city.**

***This algorithm can be summarized by the following steps:***

1. **Create a set of all cities (U) and an ordered set of cities representing a tour (V).**
2. **Choose arbitrary starting city from V, remove from V and add to U.**
3. **Select the city in V closest to the latest city added to U, remove from V and add to U.**
4. **Repeat step 2 until no cities remain in V.**

CODE

#include<iostream>

using namespace std;

#define INT\_MAX 999999;

int n=4 , dist[10][10];

void takeinput()

{

cout<<"Number of Cities = 4"<<endl;

int i,j;

cout<<"enter the distance between cities: ";

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

cin>>dist[i][j];

}

}

}

int visited\_all = (1<<n) - 1;

//if all cities have been visited

int tsp(int mask, int pos )

{

if(mask==visited\_all)

{

return dist[pos][0];

}

//try to go to unvisited cities

int ans= INT\_MAX;

for(int city=0;city<n;city++)

{

if((mask&(1<<city))==0)

{

int newans= dist[pos][city] + tsp( mask|(1<<city),city);

ans= min(ans,newans);

}

}

return ans;

}

int main()

{

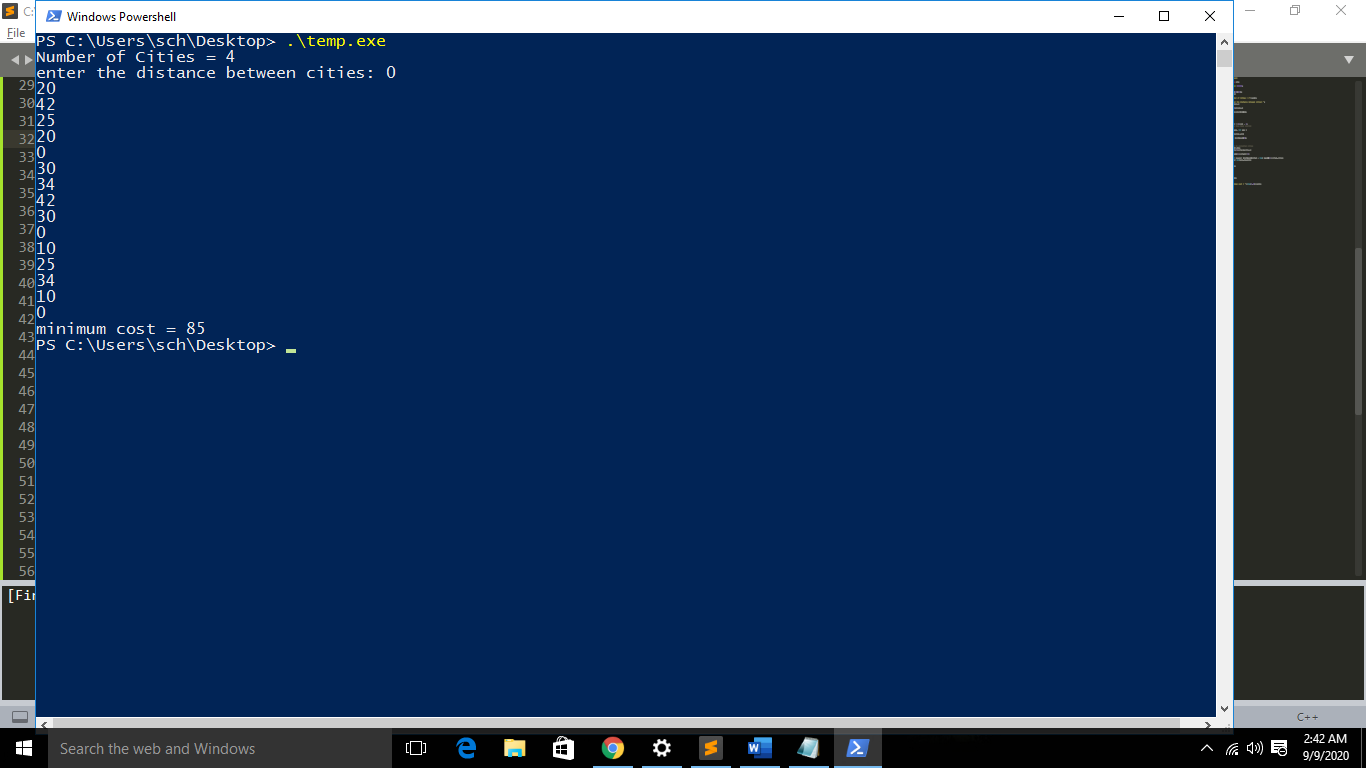
takeinput();

cout<<"minimum cost = "<<tsp(1,0)<<endl;

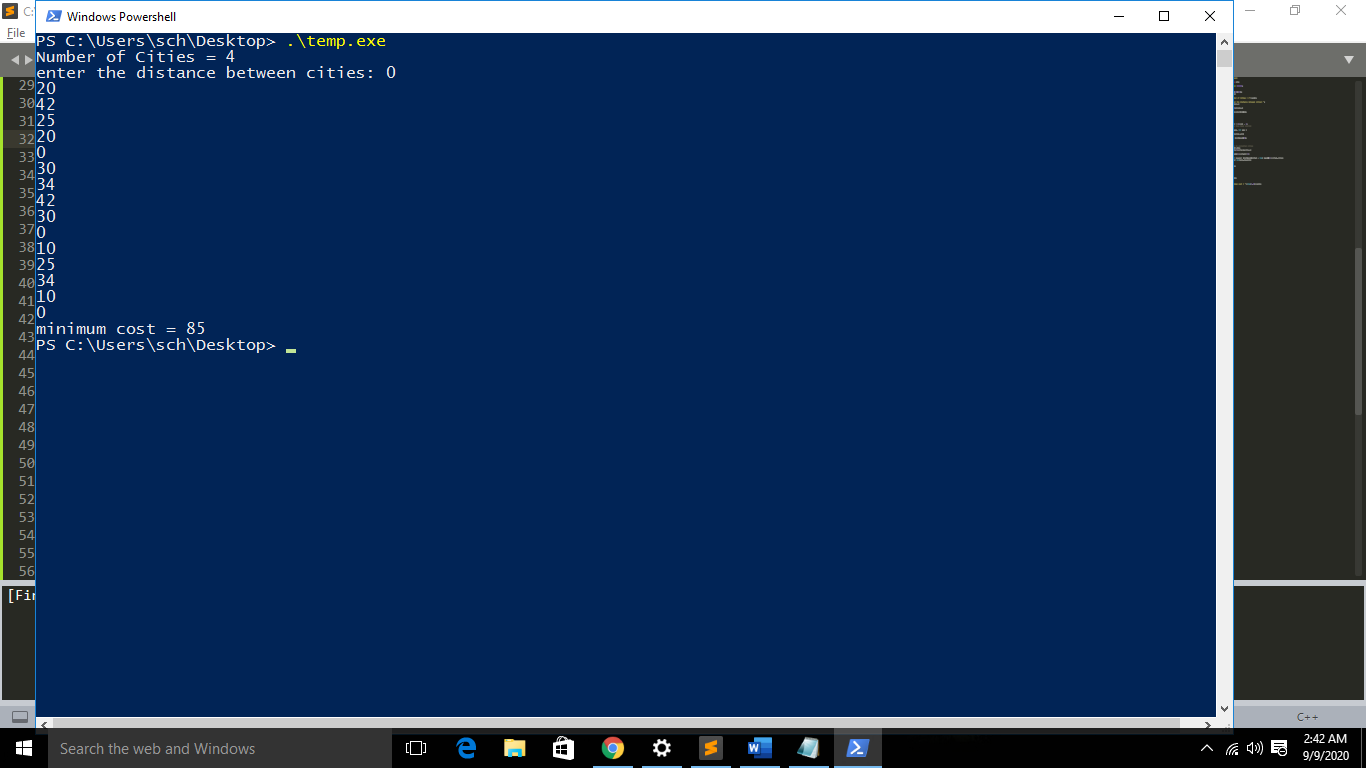
return 0;

}

INPUT



OUTPUT



Minimum cost = 85