**Experiment No: 8**

**Name:Shubham Kaspate**

**Roll No:20141241**

**Batch:I3**

**//Implement Multistage Graph using DP**

#include<bits/stdc++.h>

using namespace std;

#define N 8

#define INF INT\_MAX

int shortestDist(int graph[N][N])

{

int dist[N];

dist[N-1] = 0;

for (int i = N-2 ; i >= 0 ; i--)

{

dist[i] = INF;

for (int j = i ; j < N ; j++)

{

if (graph[i][j] == INF)

continue;

dist[i] = min(dist[i], graph[i][j] +dist[j]);

}

}

cout<<"minimum distance between from source node to target node:"<<endl;

return dist[0];

}

int main()

{

int graph[N][N] =

{{INF, 1, 5, 7, INF, INF, INF, INF},

{INF, INF, INF, INF, 3, 13, INF, INF},

{INF, INF, INF, INF, 7, 8, 26, INF},

{INF, INF, INF, INF, INF, INF, 2, INF},

{INF, INF, INF, INF, INF, INF, INF, 9},

{INF, INF, INF, INF, INF, INF, INF, 3},

{INF, INF, INF, INF, INF, INF, INF, 2},

{INF, INF, INF, INF, INF, INF, INF, INF}};

cout << shortestDist(graph);

return 0;

}

Output:

