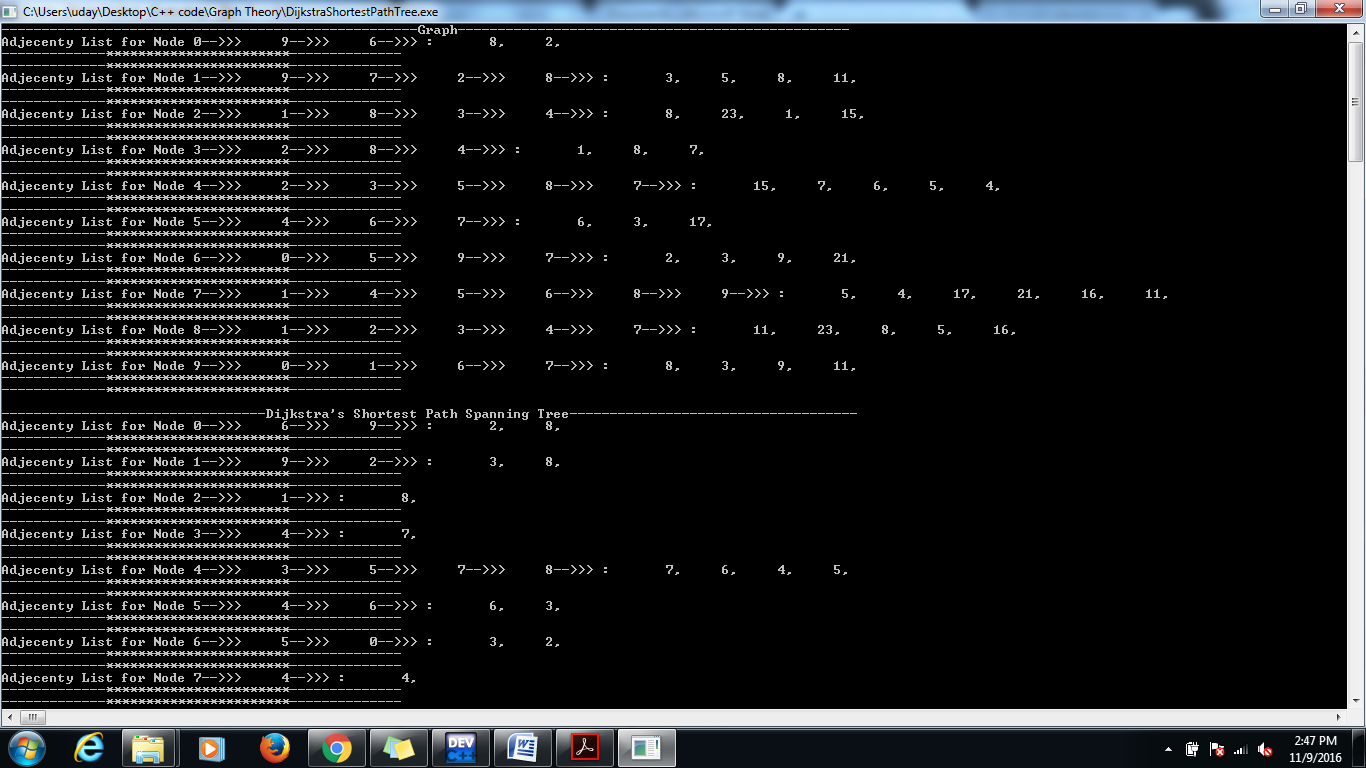
Graph specifications

The Assignment has been written in C++. The Graph used has 10 vertices and 20 edges and is explained below. I have defined the graph in terms of an adjacency list, which is also drawn out in the document below. I didn't create a separate project for this assignment and created a new .cpp file for the propose of this assignment..



The graph has been structures here and is an undirected graph. For the edge weight please refer to the adjacency list above.

Dijkstra's algorithm to find the shortest path Tree from a given node for a graph

The algorithm has been implemented with the help of a function namely DijkstraShortestPath. The function takes the node number as argument. It uses priority queue Standard library and uses 3 array

int dist[Ver];//to track the shortest distance of the node from the starting node

int child[Ver][2];//a 2-d array to track the child of the index and it's weight

bool visited[Ver]={false};//a Boolean array to track visited nodes

I have also used a myComparator class to pass to the minimum priority tree.

here is the shortest path spanning tree, for the graph explained above.

