Subject : DAA LAB			Software : Ubuntu	
			Hardware : Core i5	
Branch : CSE	Semester : 4th		Page No. 21	Prog No. 06

EM STATEMENT Write a Program to find shortest Paths to other vertices from a given vertex in a weighted connected graph using Dijkstra's algorithm.

RITHM & CODE :

```
#include <stdio.h>
# include <limits.h>
# include <stdbool.h>
# define MAX_VERTICES 100
int minDistance (int dist[], bool sptset [], int vertices)
 { int main = INT_MAX , min_index;
    for (int v=0; v< vertices; v++)
    { if (!sptset [v] && dist [v] <= min)
      { min = dist [v];
         min_index = v;
      }
    }
    return min_index;
 }
void Printsolutions (int dist[], int vertices , int src)
  { printf (" vertex \t \t Distance from Source (%d) \n",
                                              src );
     for ( int i=0; i< vertices; i++){
          printf ("%d \t \t %d \n", i, dist [i]);
     }
  }
void dijkstra (int graph [MAX_VERTICES] [MAX_VERTICES]
                  , intsrc , int vertices)
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

Signature of Faculty	Signature of Student Rudranarayan Sahoo
GRADE :	Date :

Subject : DAA LAB		Software : Ubuntu	
		Hardware : Core i5	
Branch : CSE	Semester : 4th	Page No. 22	Prog No. 06

```
{int dist [vertices];
 bool sptset [vertices];
 for (int i=0; i< vertices;i++)
 {dist [i] = INT_MAX;
  sptset [i] = false;
 }
dist [src] = 0;
 for (int count = 0; count < vertices -1; count ++)
  {int u = minDistance (dist, spt set, vertices);
   sptset [u] = true;
   for (int v=0; v< vertices ;v++)
   {if (!spt set [v] && graph [u][v] && dist [u]
      != INT_MAX && dist [u] + graph [u][v] <
      dist [v])
      {dist [v] = dist [u] + graph [u][v];
      }
   }
 }
 PrintSolutions (dist, vertices ,src);
}
int main ()
 {int vertices ,edges , src;
```

Signature of Faculty

Date :

Signature of Student Rudranarayan Sahoo

Date :

Subject : DAA LAB		Software : Ubuntu	
		Hardware : core i5	
Branch : CSE	Semester : 4th	Page No. 23	Prog No. 06

LEM STATEMENT

ORITHM & CODE :

```
int graph [MAX_VERTICES][MAX_VERTICES] = {0};
Print f ("Enter number of vertices in the graph
          (max %d):" MAX_VERTICES);
Scanf ("%d", & vertices);
Printf ("Enter number of edges in the graph:");
Scanf ("%d", &edges);
for (int i=0; i< vertices; i++)
  {for (int j=0; j< vertices ; j++)
     {graph [i][j] =0;
     }
  }
Printf(" Enter edges in the format 'UV weight'
      (vertices should be 0-based ): \n");
for (int i=0; i<edges ; i++)
    {int u,v, weight;
     Scanf ("%d %d %d", &u, &v, & weight);
     graph [u][v] = weight;
     graph [v][u] = weight;
    }
Printf ("Enter the source vertex (0 to %d): ",
          vertices - 1);
Scanf ("%d", &src);
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student Rudranarayan Sahoo

Date :

Subject : DAA LAB		Software : Ubuntu	
		Hardware : Core i5	
Branch : CSE	Semester : 4th	Page No. 24	Prog No. 06

LEM STATEMENT

ORITHM & CODE :

```
if (srcc <0 ||srcc >= vertices)
  {printf ("Invalid source vertex !\n");
   return1;
  }
dijkstra (graph, src , vertices);
return 0;
}
```

Output Obtained:

```
Enter number of vertices in the graph
(max 100): 6
Enter number of edges in the graph: 9
Enter edges in the formed UV weight (vertices
should be 0- based ):
 0  1  4
 0  2  5
 1  2  11
 1  3  9
 1  4  7
 2  4  3
 3  4  13
 3  5  2
 4  5  6
```

NPUT GIVEN

UTPUT OBTAINED

EMARKS

RADE :

Signature of Faculty

Date :

Signature of Student Rudranarayan Sahoo

Date :

Subject : DAA LAB		Software : Ubuntu Hardware : Core i5	
Branch : CSE	Semester : 4th	Page No. 25	Prog No. 06

LEM STATEMENT

ORITHM & CODE :

```
Enter the Source vertex (0 to 5): 0
Vertex          Distance from Source (0)

  0                  0
  1                  4
  2                  5
  3                  13
  4                  8
  5                  14
```

Time Complexity : $O(V^2)$

INPUT GIVEN	
OUTPUT OBTAINED	
REMARKS	

GRADE :	Signature of Faculty Date :	Signature of Student Rudranarayan Sahoo Date :