ROLL NO – 21BCM078

NAME – ZEEL VANANI

PDS

PRACTICAL – 10

Q1. Write a program to obtain a spanning tree of a connected undirected graph using appropriate data structure.

INPUT:

#include <stdio.h>

#include <limits.h>

#define vertices 5

int minimum\_key(int k[], int mst[])

{

int minimum = INT\_MAX, min,i;

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

if (mst[i] == 0 && k[i] < minimum )

minimum = k[i], min = i;

return min;

}

void prim(int g[vertices][vertices])

{

int parent[vertices];

int k[vertices];

int mst[vertices];

int i, count,edge,v;

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

{

k[i] = INT\_MAX;

mst[i] = 0;

}

k[0] = 0;

parent[0] = -1;

for (count = 0; count < vertices-1; count++)

{

edge = minimum\_key(k, mst);

mst[edge] = 1;

for (v = 0; v < vertices; v++)

{

if (g[edge][v] && mst[v] == 0 && g[edge][v] < k[v])

{

parent[v] = edge, k[v] = g[edge][v];

}

}

}

printf("\n Edge \t Weight\n");

for (i = 1; i < vertices; i++)

printf(" %d <-> %d %d \n", parent[i], i, g[i][parent[i]]);

}

int main()

{

int g[vertices][vertices] = {

{1, 0, 3, 0, 0},

{0, 0, 1, 1, 0},

{1, 2, 0, 1, 1},

{0, 1, 1, 0, 1},

{0, 0, 7, 1, 0},

};

prim(g);

return 0;

}

OUTPUT:

