Shrihari K Mangle
CSB2-74 12010517

LAB-09

9. Implement Prims and Kruskal's algorithm on given graph to generate MST.

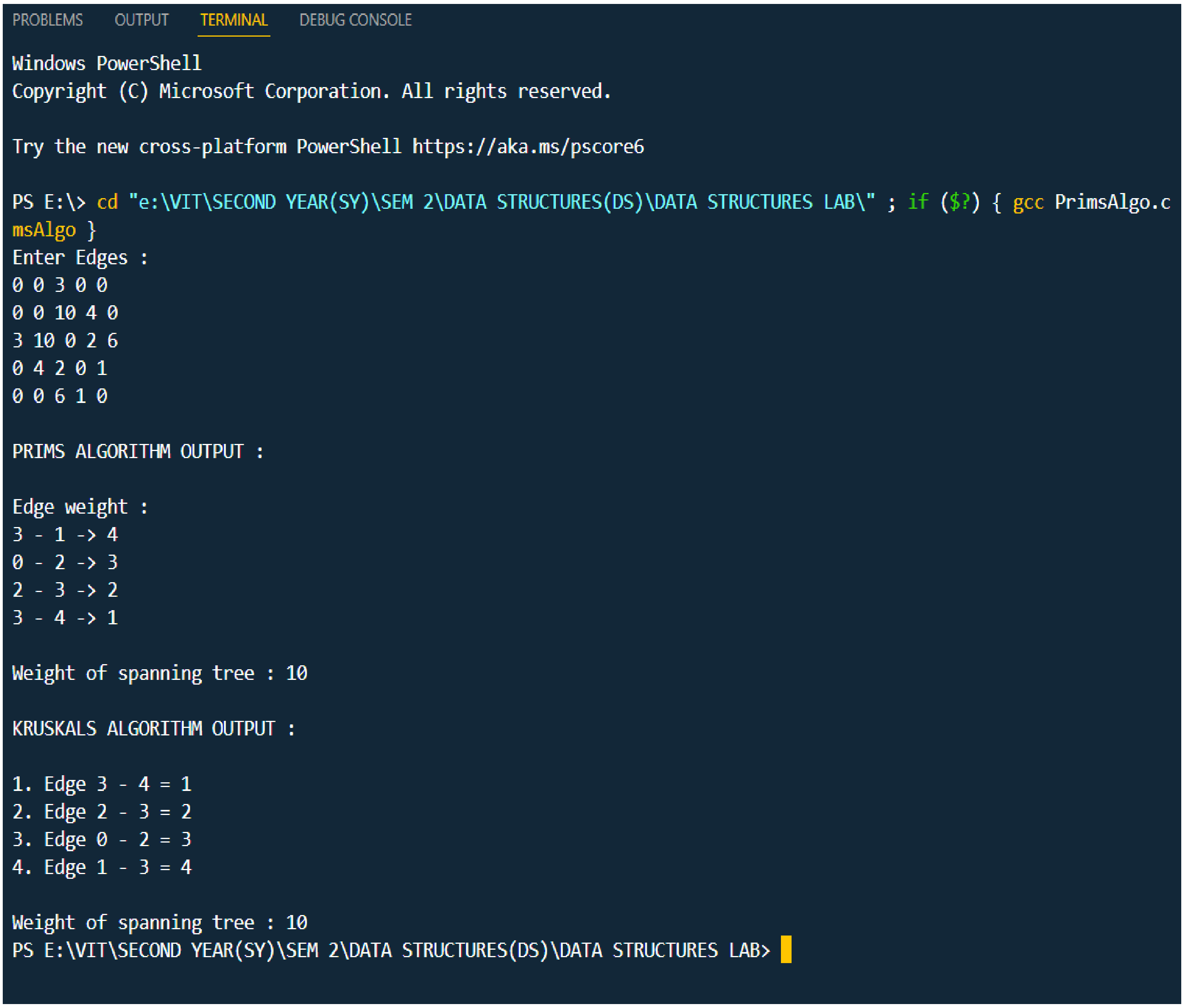
CODE :

```
#include <stdio.h>
#include <stdlib.h>

#define n 5

void displayMST(int arr[n][n], int parent[n])
{
    int sum = 0;
    printf("Edge weight : \n");

    for (int i = 1; i < n; i++)
    {
        printf("%d - %d -> %d\n", parent[i], i, arr[i][parent[i]]);
        sum += arr[i][parent[i]];
    }

    printf("\nWeight of spanning tree : %d\n", sum);
}

int closestVertex(int weight[n], int visited[n])
{
    int index, min = 999;

    for (int i = 0; i < n; i++)
    {
        if (visited[i] == 0 && weight[i] < min)
        {
            min = weight[i];
            index = i;
        }
    }
    return index;
}

void printArray(int *arr)
{
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void PrimsAlgo(int arr[n][n])
{
    int u, parent[n], weight[n], visited[n];
    for (int i = 0; i < n; i++)
    {
        weight[i] = 999;
        visited[i] = 0;
        parent[i] = 100;
    }

    parent[0] = -1;
    weight[0] = 0;

    for (int i = 0; i < n; i++)
    {
        u = closestVertex(weight, visited);

        // printf("Index of non-visited vertex having minimum weight :
%d\n", u);

        visited[u] = 1;

        // printf("v : ");
        // printArray(visited);

        for (int j = 0; j < n; j++)
        {
            if (arr[u][j] != 0 && visited[j] == 0 && arr[u][j] < weight[j])
            {
                parent[j] = u;
                weight[j] = arr[u][j];
            }
        }
        // printf("p : ");
        // printArray(parent);

        // printf("w : ");
        // printArray(weight);

        // printf("\n");
    }

    displayMST(arr, parent);
}

int parent[n];
```

```c
int findParent(int i)
{
    while (parent[i])
    {
        i = parent[i];
    }
    return i;
}

int Union(int i, int j)
{
    if (i != j)
    {
        parent[j] = i;
        return 1;
    }
    return 0;
}

void kruskalsAlgo(int arr[n][n])
{
    int min = 999, u, v, a, b, mincost = 0, x;
    int e = 1;

    while (e < n)
    {
        min = 999;
        for (int i = 0; i < n; i++)
        {

            for (int j = 0; j < n; j++)
            {
                if (arr[i][j] < min)
                {
                    min = arr[i][j];
                    a = u = i;
                    b = v = j;
                }
            }
        }

        u = findParent(u);
        v = findParent(v);

        if (Union(u, v))
        {
            printf("%d. Edge %d - %d = %d\n", e++, a, b, min);
            mincost += min;
        }

        arr[a][b] = arr[b][a] = 999;
    }

    printf("\nWeight of spanning tree : %d\n", mincost);
}

int main()
{

    printf("Enter Edges : \n");

    int adjMatrix[n][n];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &adjMatrix[i][j]);

            if (adjMatrix[i][j] == 0)
            {
                adjMatrix[i][j] = 999;
            }
        }
    }

    // int adjMatrix[n][n] = {
    //     {0, 0, 3, 0, 0},
    //     {0, 0, 10, 4, 0},
    //     {3, 10, 0, 2, 6},
    //     {0, 4, 2, 0, 1},
    //     {0, 0, 6, 1, 0},
    // };

    printf("\nPRIMS ALGORITHM OUTPUT : \n\n");
    PrimsAlgo(adjMatrix);

    printf("\nKRUSKALS ALGORITHM OUTPUT : \n\n");
    kruskalsAlgo(adjMatrix);

    return 0;
}
```

Shrihari K Mangle
CSB2-74 12010517

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\> cd "e:\VIT\SECOND YEAR(SY)\SEM 2\DATA STRUCTURES(DS)\DATA STRUCTURES LAB\" ; if ($?) { gcc PrimsAlgo.c
msAlgo }
Enter Edges :
0 0 3 0 0
0 0 10 4 0
3 10 0 2 6
0 4 2 0 1
0 0 6 1 0

PRIMS ALGORITHM OUTPUT :

Edge weight :
3 - 1 -> 4
0 - 2 -> 3
2 - 3 -> 2
3 - 4 -> 1

Weight of spanning tree : 10

KRUSKALS ALGORITHM OUTPUT :

1. Edge 3 - 4 = 1
2. Edge 2 - 3 = 2
3. Edge 0 - 2 = 3
4. Edge 1 - 3 = 4

Weight of spanning tree : 10
PS E:\VIT\SECOND YEAR(SY)\SEM 2\DATA STRUCTURES(DS)\DATA STRUCTURES LAB>
```