**Slip 6**

Q-1) Write a program for the Implementation of Prim’s algorithm to find minimum cost spanning tree.

#include<iostream>

using namespace std;

int ne=1,min\_cost=0;

int main()

{

    int n,i,j,min,cost[20][20],a,u,b,v,source, visited[20];

    cout<<"Enter the no. of nodes:";

    cin>>n;

    cout<<"Enter the cost matrix:\n";

    for(i=1;i<=n;i++)

    {

        for(j=1;j<=n;j++)

        {

            cin>>cost[i][j];

        }

    }

    for(i=1;i<=n;i++)

    visited[i]=0;

    cout<<"Enter the root node:";

    cin>>source;

    visited[source]=1;

    cout<<"\nMinimum cost spanning tree is\n";

    while(ne<n)

    {

        min-999;

        for(i=1;i<=n;i++)

        {

            for(j=1;j<=n;j++)

            {

                if(cost[i][j]<min)

                    if(visited[i]=0)

                        continue;

                    else

                    {

                        min=cost[i][j];

                        a=u=i;

                        b=v=j;

                    }

                }

            }

            if(visited[u]==0||visited[v]==0)

            {

                cout<<"\nEdge"<<ne++<<"\t("<<a<<"->"<<b<<")="<<min;

                min\_cost-min\_cost+min;

                visited[b]=1;

            }

            cost[a][b]=cost[b][a]=999;

        }

    cout<<"\nMinimum cost="<<min\_cost<<"\n";

}

Q.2) Write a Program to find only length of Longest Common Subsequence

// A Naive recursive implementation

// of LCS problem

#include <stdio.h>

#include<string.h>

int max(int a, int b);

// Returns length of LCS for X[0..m-1],

// Y[0..n-1]

int lcs(char\* X, char\* Y, int i, int j)

{

    if (X[i] == 0 || Y[j] == 0)

        return 0;

    if (X[i] == Y[j])

        return 1 + lcs(X, Y, i + 1, j + 1);

    else

        return max(lcs(X, Y, i, j + 1),

                lcs(X, Y, i + 1, j));

}

// Utility function to get max of

// 2 integers

int max(int a, int b) { return (a > b) ? a : b; }

// Driver code

int main()

{

    char S1[] = "BD";

    char S2[] = "ABCD";

    int m = strlen(S1);

    int n = strlen(S2);

    int i = 0, j = 0;

    // Function Call

    printf("Length of LCS is %d", lcs(S1, S2, i, j));

    return 0;

}