Name: Surwade Trisharan Rajesh

Roll no.: 48

//write a program to find the shortest path using single source pair shortest path

```
#include<iostream>
#include<conio.h>
using namespace std;
class shortest
{
private:
    int n, cost[20][20];
public:
    void getdata();
    void shortestpath(int v);

};
void shortest::getdata()
{
    cout << "Enter the number of the vertices:\n";
    cin >> n;
    cout << "\nEnter the Adjacent Matrix=\n";
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            cin >> cost[i][j];
        }
    }
}
void shortest::shortestpath(int v)
{
    int s[50], dist[50], i, j, d1, d2, u;
    for (i = 1; i <= n; i++)
    {
        s[i] = 0;
        dist[i] = cost[v][i];
    }
    s[v] = 1;
    dist[v] = 0;
    for (int num = 2; num <= n - 1; num++)
    {
        int min = 999;
        for (int i = 1; i <= n; i++)
        {
            if (dist[i] < min && s[i == 0])
            {
                u = i;
                min = dist[i];
            }

        }
        s[u] = 1;
        for (int j = 1; j <= n;j++)
        {
            if (s[j] = 0)
            {
                d1 = dist[j];
                d2 = dist[u] + cost[u][j];
                dist[j] = d1 < d2 ? d1 : d2;

            }
        }
    }
    for (int i = 1; i <= n; i++)
    {
        cout << "\n Distance of vertex " << v << " from vertex " << i << " is " <<
dist[i];
    }

}
int main()
{
    int v, i;
    shortest s;
    s.getdata();
    cout << " Enter the starting vertex : \n";
    cin >> v;
    s.shortestpath(v);
    return 0;
}
```

Output:

```
Enter the number of the vertices:

3


Enter the Adjacent Matrix=

2 3 4

5 6 7

8 9 1

 Enter the starting vertex :

1


 Distance of vertex 1 from vertex 1 is 0

 Distance of vertex 1 from vertex 2 is 3

 Distance of vertex 1 from vertex 3 is 4
```