B. Sai Hithesh

1BM17CS021

Dijkstra's Algorithm

```
#include <iostream>
using name space std;
int a[30][30], source, a[30], p[30];
void alg [int a[][30], int n]
{
    ~~int~~ in + s[n]
    for (int i=0; i<n; i++)
    {
        d[i] = a[source][i];
        p[i] = source;
        s[i] = 0;
    }
    s[source] = 1
    for (int c=0; c<n; c++)
    {
        int min = 9999, v;
        for (int j=0; j<n; j++)
        {
            if (d[j] < min && s[j]!=1)
            {
                min = d[j];
                v = j;
            }
        }
```

B. Saitlithesh
IBM17CS021

```
        S[v] = 1;
        for (int i=0; i<n; i++)
        {
            if (min + a[v][i] < d[i])
            {
                d[i] = min + a[v][i];
                P[i] = v;
            }
        }
    }
}
int main() {
    int n;
    cout << "Enter no of vertices" << endl;
    cin >> n;
    cout << "Enter the graph" << endl;
    for (int i=0; i<n; i++)
    {
        cin >> a[i][j];
    }
    cout << "Enter the source vertex" << endl;
    cin >> source;
    cout << "The shortest paths from vetex "
         << source << " are :" << endl;
```

```
alg(a,n);
for (int i=0; i<n; i++){
    int k=i;
    while (k!= source) {
        Count << k << " <- ";
        k = p[k];
    }
    Cout << source << " = ";
    Cout << d[i] << endl;
}
return 0;
}
```