

Assignment PRIMS ALGORITHM

```
#include <iostream>

#define I 32767 // Represents infinity for no direct connection

using namespace std;

void PrintMST(int T[][20], int G[][20], int V) {
    cout << "\nMinimum Spanning Tree Edges (w/ cost)\n" << endl;
    int sum {0};
    for (int i {0}; i < V - 1; i++) {
        int c = G[T[0][i]][T[1][i]];
        cout << "[" << T[0][i] << "]-[" << T[1][i] << "] cost: " << c << endl;
        sum += c;
    }
    cout << endl;
    cout << "Total cost of MST: " << sum << endl;
}

void PrimsMST(int G[][20], int V) {
    int u, v, min = I;
    int track[20], T[2][20] {0};

    // Initial step: Find the minimum cost edge
    for (int i = 0; i < V; i++) {
        for (int j = i; j < V; j++) {
            if (G[i][j] < min && G[i][j] != 0) {
                min = G[i][j];
                u = i;
                v = j;
            }
        }
    }
}
```

```
    }  
    }  
}
```

```
T[0][0] = u;
```

```
T[1][0] = v;
```

```
track[u] = track[v] = 0; // Mark selected nodes
```

```
// Initialize the track array
```

```
for (int i = 0; i < V; i++) {  
    if (track[i] != 0) {  
        track[i] = (G[i][u] < G[i][v]) ? u : v;  
    }  
}
```

```
// Main loop
```

```
for (int i = 1; i < V - 1; i++) {  
    int k;  
    min = l;  
    for (int j = 0; j < V; j++) {  
        if (track[j] != 0 && G[j][track[j]] < min) {  
            k = j;  
            min = G[j][track[j]];  
        }  
    }  
    T[0][i] = k;  
    T[1][i] = track[k];  
    track[k] = 0;
```

```
// Update the track array
```

```
for (int j = 0; j < V; j++) {
```

```

        if (track[j] != 0 && G[j][k] < G[j][track[j]]) {
            track[j] = k;
        }
    }
}

PrintMST(T, G, V);
}

int main() {
    int V;

    cout << "Enter the number of vertices: ";
    cin >> V;

    int G[20][20];
    cout << "Enter the adjacency matrix (" << V << "x" << V << "):\n";
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            cin >> G[i][j];
            if (G[i][j] == 0 && i != j) {
                G[i][j] = 1; // Replace 0 with infinity for no direct connection
            }
        }
    }

    PrimsMST(G, V);

    return 0;
}

```