```
#include <iostream>
#define INF 9999
using namespace std;

void primMST(int graph[10][10], int n) {
   int selected[10];    // Tracks selected vertices
   int no_edge = 0;     // Count of edges in MST
   int totalCost = 0;   // Total cost of MST

   for (int i = 0; i < n; i++)
      selected[i] = 0;

   selected[0] = 1;  // Start from vertex 0

   cout << "Edge \tWeight\n";

   while (no_edge < n - 1) {
      int min = INF, x = 0, y = 0;

      for (int i = 0; i < n; i++) {
         if (selected[i]) {
            for (int j = 0; j < n; j++) {
               if (!selected[j] && graph[i][j]) {
                  if (min > graph[i][j]) {
                     min = graph[i][j];
                     x = i;
```

```
                    y = j;
                }
            }
        }
    }
}

    cout << x << " - " << y << " \t" << graph[x][y] << endl;
    totalCost += graph[x][y];
    selected[y] = 1;
    no_edge++;
  }

  cout << "Total cost of MST: " << totalCost << endl;
}

// -------- Main Function --------
int main() {
  int n;
  int graph[10][10];

  cout << "Enter number of vertices: ";
  cin >> n;

  cout << "Enter adjacency matrix (0 for no edge):\n";
  for (int i = 0; i < n; i++)
```

```
        for (int j = 0; j < n; j++) {
            cin >> graph[i][j];
            if (graph[i][j] == 0)
                graph[i][j] = INF;
        }

    primMST(graph, n);

    return 0;
}
```

```
#include <iostream>
using namespace std;

#define V 4  // Number of vertices

void primMST(int graph[V][V]) {
    int selected[V] = {0};
    selected[0] = 1; // Start from node 0

    cout << "Prim's MST edges:\n";

    int edgeCount = 0;
    while (edgeCount < V - 1) {
```

```
    int min = 9999;
    int u = 0, v = 0;

    for (int i = 0; i < V; i++) {
      if (selected[i]) {
        for (int j = 0; j < V; j++) {
          if (!selected[j] && graph[i][j]) {
            if (graph[i][j] < min) {
              min = graph[i][j];
              u = i;
              v = j;
            }
          }
        }
      }

    cout << u << " - " << v << " : " << graph[u][v] << endl;
    selected[v] = 1;
    edgeCount++;
  }
}

int main() {
  int graph[V][V] = {
    {0, 2, 0, 6},
```

```
        {2, 0, 3, 8},
        {0, 3, 0, 0},
        {6, 8, 0, 0}
    };

    primMST(graph);
    return 0;
}
```