```
import java.util.*;

public class Prim {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of vertices");
        int n = sc.nextInt(), minCost = 0, edges = 0;
        int[][] cost = new int[n][n];
        boolean[] visited = new boolean[n];

        System.out.println("Enter the cost matrix");
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++) {
                cost[i][j] = sc.nextInt();
                if (cost[i][j] == 0 && i != j) cost[i][j] = 999; // No edge
            }

        visited[0] = true;
        PriorityQueue<int[]> pq = new PriorityQueue<>(Comparator.comparingInt(a -> a[1]));
        for (int j = 1; j < n; j++)
            if (cost[0][j] != 999) pq.add(new int[]{j, cost[0][j]});

        while (edges < n - 1 && !pq.isEmpty()) {
            int[] edge = pq.poll(), u = edge[0], weight = edge[1];
            if (!visited[u]) {
                visited[u] = true; minCost += weight; edges++;
                System.out.println((edges) + " edge(0," + u + ") = " + weight);
                for (int j = 0; j < n; j++)
                    if (!visited[j] && cost[u][j] != 999) pq.add(new int[]{j, cost[u][j]});
            }
        }
        System.out.println("The minimum cost of spanning tree is " + minCost);
    }
}
```

Lab 8