```
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#define V 5
int minKey(int key[], bool mstSet[]) {
  int min = INT_MAX, min_index;
   for (int v = 0; v < V; v++)
      if (mstSet[v] == false && key[v] < min)
         min = key[v], min_index = v;
   return min_index;
}
int printMST(int parent[], int graph[V][V]) {
   printf("Edge \tWeight\n");
   for (int i = 1; i < V; i++)
      printf("%d - %d \t%d \n", parent[i], i, graph[i][parent[i]]);
}
void primMST(int graph[V][V]) {
   int parent[V];
   int key[V];
   bool mstSet[V];
   for (int i = 0; i < V; i++)
      key[i] = INT_MAX, mstSet[i] = false;
   key[0] = 0;
   parent[0] = -1;
   for (int count = 0; count < V - 1; count++) {
      int u = minKey(key, mstSet);
      mstSet[u] = true;
 for (int v = 0; v < V; v++)
  if (graph[u][v] && mstSet[v] == false && graph[u][v] < key[v])
           parent[v] = u, key[v] = graph[u][v];
   } printMST(parent, graph);
}
int main() {
   int graph[V][V] = { { 0, 2, 0, 6, 0 },   { 2, 0, 3, 8, 5 }, { 0, 3, 0, 0, 7 },{ 6, 8, 0, 0, 9 },
                { 0, 5, 7, 9, 0 } };
primMST(graph);
   return 0;}
```

OUTPUT

```
Edge     Weight
  0 - 1     2
  1 - 2     3
  0 - 3     6
  1 - 4     5
```

```
import java.io.*;
class GFG {
   public int[][] adjacencyMatToIncidenceMat(int[][] adj) {
      int vertices = adj.length, edges = 0;
      for (int i = 0; i < adj.length; i++) {
         for (int j = i + 1; j < adj[i].length; j++) {
            if (adj[i][j] > 0)
               edges++;} }
      int[][] incidenceMat = new int[vertices][edges];
      for (int i = 0; i < vertices; i++) {
         for (int j = i + 1; j < adj[i].length; j++) {
            int edgeNumber = adj[i][j];
            if (edgeNumber > 0) {
               incidenceMat[i][edgeNumber - 1] = 1;
               incidenceMat[j][edgeNumber - 1] = 1;}}}
      return incidenceMat;}
 public static void main(String[] args) {

      GFG gfg = new GFG();
 int[][] adj = {
         { 0, 1, 0, 4 },
         { 1, 0, 2, 0 },
         { 0, 2, 0, 3 },
         { 4, 0, 3, 0 }   };
      int[][] incidence = gfg.adjacencyMatToIncidenceMat(adj);
      for (int[] row : incidence) {
         for (int val : row) {
            System.out.print(val);}
System.out.println();}}}
```