**#include <stdio.h>**

**#include <limits.h>**

**int minKey(int key[], int mstSet[], int V) {**

**int min = INT\_MAX, min\_index;**

**for (int v = 0; v < V; v++) {**

**if (mstSet[v] == 0 && key[v] < min) {**

**min = key[v];**

**min\_index = v;**

**}**

**}**

**return min\_index;**

**}**

**// Function to print the constructed MST stored in parent[]**

**void printMST(int parent[], int \*\*graph,int V) {**

**printf("Edge Weight\n");**

**int minimumweight=0;**

**for (int i = 1; i < V; i++){**

**printf("%d - %d %d \n", parent[i], i, i, parent[i], graph[i][parent[i]]);**

**minimumweight=minimumweight+graph[i][parent[i]];**

**}**

**printf("Minimumweight is %d",minimumweight);**

**}**

**// Function to construct and print MST for a graph represented using adjacency matrix representation**

**void primMST(int \*\*graph,int V) {**

**int parent[V]; // Array to store constructed MST**

**int key[V]; // Key values used to pick minimum weight edge in cut**

**int mstSet[V];**

**int i; // To represent set of vertices not yet included in MST**

**int j;**

**// Initialize all keys as INFINITE**

**for (int i = 0; i < V; i++) {**

**key[i] = INT\_MAX;**

**mstSet[i] = 0;**

**}**

**// Always include first vertex in MST.**

**// Make key 0 so that this vertex is picked as first vertex.**

**key[0] = 0; // First vertex is always root of MST**

**parent[0] = -1; // First node is always root of MST**

**// The MST will have V vertices**

**for (int count = 0; count < V - 1; count++) {**

**// Pick the minimum key vertex from the set of vertices not yet included in MST**

**int u = minKey(key, mstSet, V);**

**// Add the picked vertex to the MST set**

**mstSet[u] = 1;**

**// Update key value and parent index of the adjacent vertices of the picked vertex.**

**// Consider only those vertices which are not yet included in MST**

**for (int v = 0; v < V; v++) {**

**// graph[u][v] is non-zero only for adjacent vertices of m**

**// mstSet[v] is false for vertices not yet included in MST**

**// Update the key only if graph[u][v] is smaller than key[v]**

**if (graph[u][v] && mstSet[v] == 0 && graph[u][v] < key[v]) {**

**parent[v] = u;**

**key[v] = graph[u][v];**

**}**

**}**

**}**

**// Print the constructed MST**

**printMST(parent, graph,V);**

**}**

**// Driver program to test above functions**

**int main(){**

**printf("Enter the number of nodes");**

**int n;**

**scanf("%d",&n);**

**printf("Enter the cost adjacency matrix");**

**int i;**

**int j;**

**int \*\*costadjacencymatrix=malloc(n\*sizeof(int\*));**

**for(i=0;i<n;i++){**

**costadjacencymatrix[i]=malloc(n\*sizeof(int));**

**for(j=0;j<n;j++){**

**scanf("%d",&costadjacencymatrix[i][j]);**

**}**

**}**

**primMST(costadjacencymatrix, n);**

**}**

