Minimum spanning tree using prims algorithm

```
#include<stdio.h>
#define SIZE 20

#define INFINITY 32767

/*This function finds the minimal spanning tree by Prim's Algorithm */

void Prim(int G[][SIZE], int nodes)

{ int tree[SIZE], i, j, k; int min_dist, v1,

v2,total=0; // Initialize the selected vertices list

for (i=0; i<nodes; i++) tree[i] = 0; printf("\n\n

The Minimal Spanning Tree Is :\n"); tree[0] =

1; for (k=1; k<=nodes-1; k++)

{ min_dist =

INFINITY;

//initially assign minimum dist as infinity

for (i=0; i<=nodes-1; i++)

{ for (j=0; j<=nodes-1;

j++)

{
if (G[i][j] && ((tree[i] && !tree[j]) || (!tree[i] && tree[j])))

{ if (G[i][j]

<min_dist)

{

min_dist=G[i][j];
v1 = i;
v2 = j;
}
}
} } printf("\n Edge (%d %d ) and weight =
%d",v1,v2,min_dist); tree[v1] = tree [v2] = 1; total =
total+min_dist;
} printf("\n\n\t Total Path Length Is =
%d",total);
} void main() { int
G[SIZE][SIZE], nodes;
int v1, v2, length, i, j, n;

printf("\n\t Prim'S Algorithm\n");
printf("\n Enter Number of Nodes in The Graph ");
scanf("%d",&nodes); printf("\n Enter Number of
Edges in The Graph "); scanf("%d",&n); for (i=0;
i<nodes; i++) // Initialize the graph for (j=0;
j<nodes; j++) G[i][j] = 0; //entering weighted
graph printf("\n Enter edges and weights \n"); for
(i=0; i<n; i++) { printf("\n Enter Edge by V1 and
V2 :"); printf("[Read the graph from starting node
```

```
0]"); scanf("%d %d", &v1,&v2); printf("\n Enter

corresponding weight :"); scanf("%d", &length);

G[v1][v2] = G[v2][v1] = length;

}


printf("\n\t");


Prim(G,nodes);
}
```

Output :

```
Input the number of vertices: 3
Input the adjacency matrix for the
graph: 1 4 3
6 3 0
9 7 0
Edge Weight
0 - 1 6
0 - 2 9
```