```
import sys


class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                                for row in range(vertices)]


    def printMST(self, parent):
        print("Edge \tWeight")
        for i in range(1, self.V):
            print(parent[i], "-", i, "\t", self.graph[i][parent[i]])


    def minKey(self, key, mstSet):


        min = sys.maxsize

        for v in range(self.V):
            if key[v] < min and mstSet[v] == False:
                min = key[v]
                min_index = v

        return min_index

    def primMST(self):


        key = [sys.maxsize] * self.V
        parent = [None] * self.V

        key[0] = 0
        mstSet = [False] * self.V

        parent[0] = -1

        for cout in range(self.V):


            u = self.minKey(key, mstSet)


            mstSet[u] = True


            for v in range(self.V):

                if self.graph[u][v] > 0 and mstSet[v] == False \
                and key[v] > self.graph[u][v]:
                    key[v] = self.graph[u][v]
                    parent[v] = u

        self.printMST(parent)


if __name__ == '__main__':


    n = int(input("Enter the number of nodes in the graph: "))
    g = Graph(n)


    for i in range(n):
        for j in range(n):
            print("If there exists a edge between",i,j,"enter weight else enter 0")
            ele = int(input())
            g.graph[i][j]=ele

    g.primMST()
```

OUTPUT:-

```
student@student:~$ python3 prims.py
Enter the number of nodes in the graph: 5
If there exists a edge between 0 0 enter weight else enter 0
0
If there exists a edge between 0 1 enter weight else enter 0
2
If there exists a edge between 0 2 enter weight else enter 0
0
If there exists a edge between 0 3 enter weight else enter 0
6
If there exists a edge between 0 4 enter weight else enter 0
0
If there exists a edge between 1 0 enter weight else enter 0
2
If there exists a edge between 1 1 enter weight else enter 0
0
If there exists a edge between 1 2 enter weight else enter 0
3
If there exists a edge between 1 3 enter weight else enter 0
8
If there exists a edge between 1 4 enter weight else enter 0
5
If there exists a edge between 2 0 enter weight else enter 0
0
If there exists a edge between 2 1 enter weight else enter 0
3
If there exists a edge between 2 2 enter weight else enter 0
0
```

```
If there exists a edge between 2 3 enter weight else enter 0
0
If there exists a edge between 2 4 enter weight else enter 0
7
If there exists a edge between 3 0 enter weight else enter 0
6
If there exists a edge between 3 1 enter weight else enter 0
8
If there exists a edge between 3 2 enter weight else enter 0
0
If there exists a edge between 3 3 enter weight else enter 0
0
If there exists a edge between 3 4 enter weight else enter 0
9
If there exists a edge between 4 0 enter weight else enter 0
0
If there exists a edge between 4 1 enter weight else enter 0
5
If there exists a edge between 4 2 enter weight else enter 0
7
If there exists a edge between 4 3 enter weight else enter 0
9
If there exists a edge between 4 4 enter weight else enter 0
0
Edge   Weight
0 - 1    2
1 - 2    3
0 - 3    6
1 - 4    5
```