

--	--	--	--	--	--	--	--

```
def main():  
    matrix = []  
    print("Enter number of nodes")  
    n = int(input())  
    print("Enter the adjacency matrix");  
    for i in range(n):  
        m = list(map(int, input().split(" ")))  
        matrix.append(m)  
    g = graph(n)  
    for i in range(n):  
        for j in range(n):  
            if matrix[i][j] == 1:  
                g.addEdge(i, j, 1)  
    for _ in range(n):  
        g.ShortestPath(-)  
        g.ShortestPath(-)  
class graph:  
    def __init__(self, n):  
        self.matrix = []  
        self.n = n  
    def addEdge(self, u, v, w):  
        self.matrix.append((u, v, w))
```



```
def printAxe (self, dist, src):
    print ("Vector Table of { } ".format
    (chr(ord('A') + src)))
    for i in range (self.n)
        print (" {0} | {1} ".format
        (chr(ord('A') + i), dist[i]))
```

```
def shortestpath (self, src):
    dist = [99] * self.n
    dist[src] = 0.
```

```
    for i in range (self.n + 1)
        for u, v, w in self.matrix:
            if dist[u] != 99 and dist[u]
            + w < dist[v]:
                dist[v] = dist[u] + w
```

```
    self.printAxe (dist, src)
```