Atharv Arya
1BM18CS020

```
Class Graph:
 def _init_(self, n):
   Self. matrix =[]
   self.n =n

 def add Edge (self, u, v, w):
    self.matrix append ((u, v, w))

 def printArr (self, dist, src):
   Print ("vector Table of {}", format (chr(ord('A')+src)))
 for i in range (self, n):
    Print ("{0}\t{1}". format (chr ( ord ('A') + i), dist[i]))

 def BellmanFord (self, src):

   dist =[99] * self.n
   dist [src] = 0

 for _ 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.Print-Arr(dist, src)
```

```
Matrix =[]
Print ("Enter no. of nodes")
 n = int (input())
 Print ("Enter adjacency Matrix")
 for i in range(n):
    g = list (map (int, input().split (" ")))
     Matrix.append(g)
x = Graph(n)
for i in range(n):
 for j in range(n):
    if Matrix [i][j] = =1;
          x. addEdge (i, j, 1)

for _ in range(n):
   x.path cal (_)
```

[illegible]