Jishala
1BM18CS118

Lab-9.

Distance vector Algorithm

```
class Routing_table:
    def _init_(Self, Current):
        Self. Current = Current
        Self.table = {'Dist': [Current], 'Cost': [0],
                      'NextH': [Current]}
        Self. neigh=[]

    def add_direct_connection(Self, P, Cost):
        Self.table['Dist'].append(P)
        Self.table['Cost'].append(cost)
        Self.table['NextH'].append(P)
        Self.neig.append(P)

    def recieve_msg(self, dest, Cost, Sender).
        flag=0
        index_of_Sender=Self.table["Dest"].index(Sender)
        Cost_to_Sender=Self.table["Cost"][index_of_Sender]
        for d, c in zip(dest, Cost:
            if d not in Self.table["Dest"]:
                Self.table["Dest"].append(d)
                Self.table["Cost"].append( Cost_to_Sender
                                           + C)
                Self.table["NextH"].append(Sender)
```

```
flag=1
elif cost_to_sender+ c< self.table["cost"]
    self.table ['cost'] (self.table ['Dest'].
    [self.table
     index(d)] = cost_to_sender+c
    self.table ["next H"] (self.table ['Dest'].
      index(d)]=Sender
      flag =1
    if flag==1:
    return self neig ,[self.table ['Dest'],
     self.table ["cost"], self.current]
    return[], []
def print_routing_table (self):
    for x,val in self.table.items()
      print(x,val)
```