CN ASSIGNMENT 3 - DVR

Name: Khushi Dave
Roll no: BT20CSE031

- Distance Vector Routing Protocol has been implemented using python threads as routers, shared queue to transfer data between threads.
- Each router maintains lists of adjacent routers, it's routing table.
- Periodically routers exchange information, re-calculate their costs with Bellman-Ford algorithm and print the routing table.
- It determines the best route between routers for data packets based on distance. It measures the distance by the number of routers a packet has to pass, one router counts as one hop.

The computation is done using distance vector algorithm :

- The Distance vector algorithm is iterative, asynchronous and distributed, to estimate minimum distance cost between every router in the network using Bellman-Ford equation.
- Distributed: It is distributed in that each node receives information from one or more of its directly attached neighbours, performs calculation and then distributes the result back to its neighbours.
- Iterative: It is iterative in that its process continues until no more information is available to be exchanged between neighbours.
- Asynchronous: It does not require that all of its nodes operate in the lock step with each other.
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as Vector

Bellman Ford Equation :

Let's say,
x : Current router
Dx(y) = Estimate of least cost from router x to router y
C(x,v) = Cost of router x to each neighbour router v
Dx = [Dx(y) : y ∈ N]= Distance vector of router x

In some way, for every adjacent router v, x has access to Dv.
We do it using SharedBuffer in distributed systems.
Estimates of least cost from router x to destination router y, via a router v, is computed using the B-F equation.

B-F equation :
Dx(y) = min { C(x,v) + Dv(y) , Dx(y) } for each router y ∈ N

Built With

- python
- threading
- time
- math
- sys
- queue
- copy

Requirements:-

- Install python
  Then
- Pip install threading time math sys queue copy

How To Run

Open terminal in code's directory,
and type
python BT20CSE031_dvr.py input1.txt

- To run test cases of input1.txt
- Similarly for rest of test files

Working of the code:

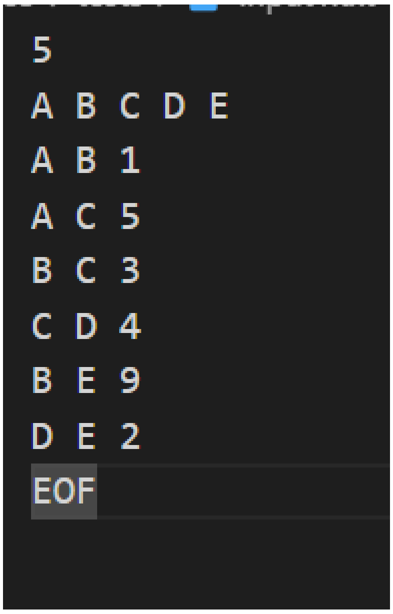
- first in Main It is Reading the input file line by line and displaying initial table
- Creates a separate thread for each router and passes it's personal routing information and shared data structure.
- Then Routing tables are sent to each neighbour after the queue lock.
- Once all tables are shared among their neighbours, next iteration tables are recomputed using Bellman-Ford equation.
- The updated tables are printed and threads wait for 2 seconds for next iteration.

Testing

Various test files were used to check:

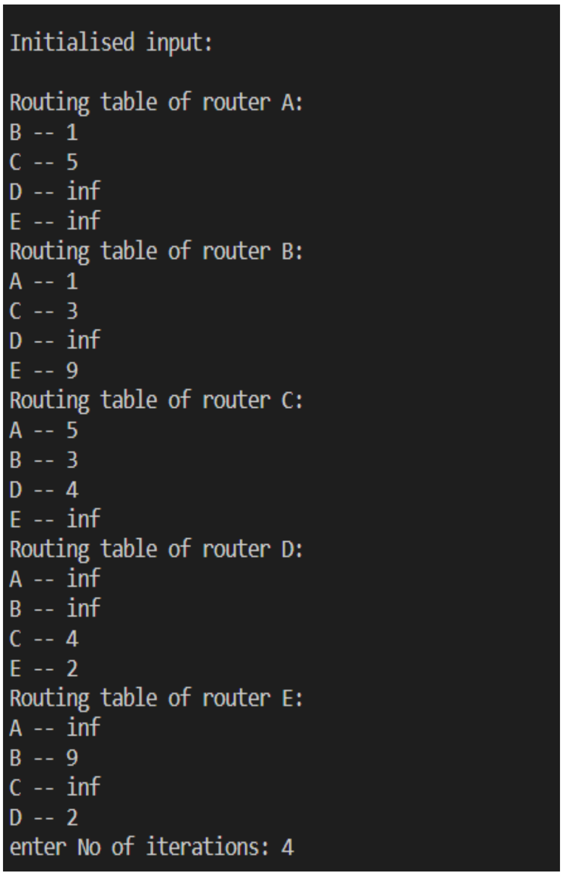
- accuracy
- concurrency

Input1.txt

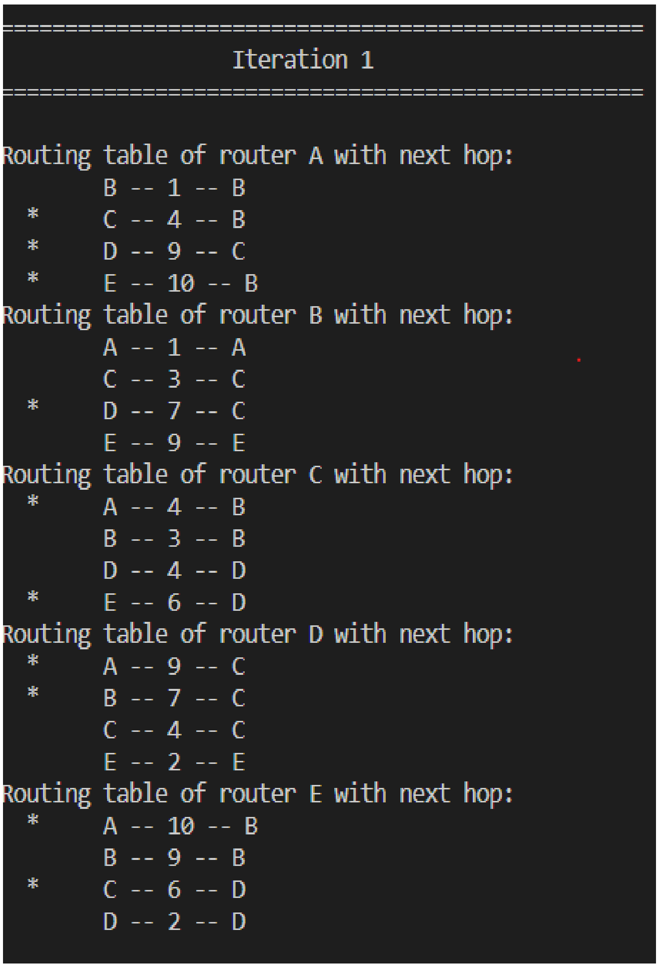


```
5
A B C D E
A B 1
A C 5
B C 3
C D 4
B E 9
D E 2
EOF
```

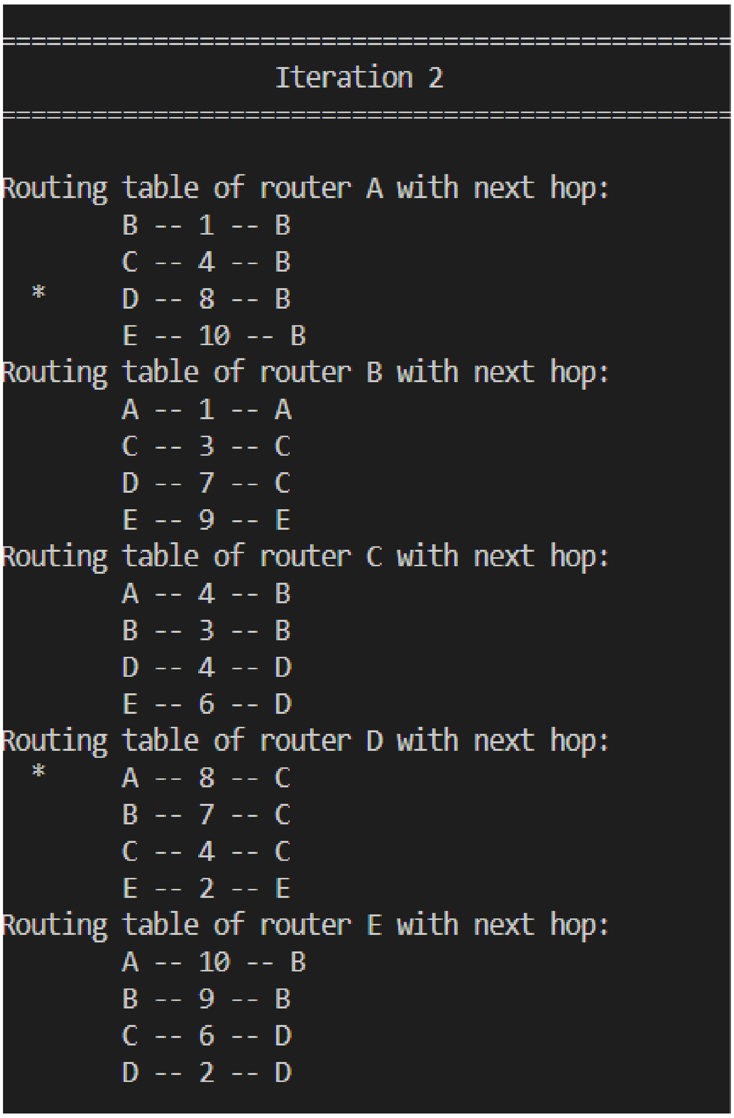
Output



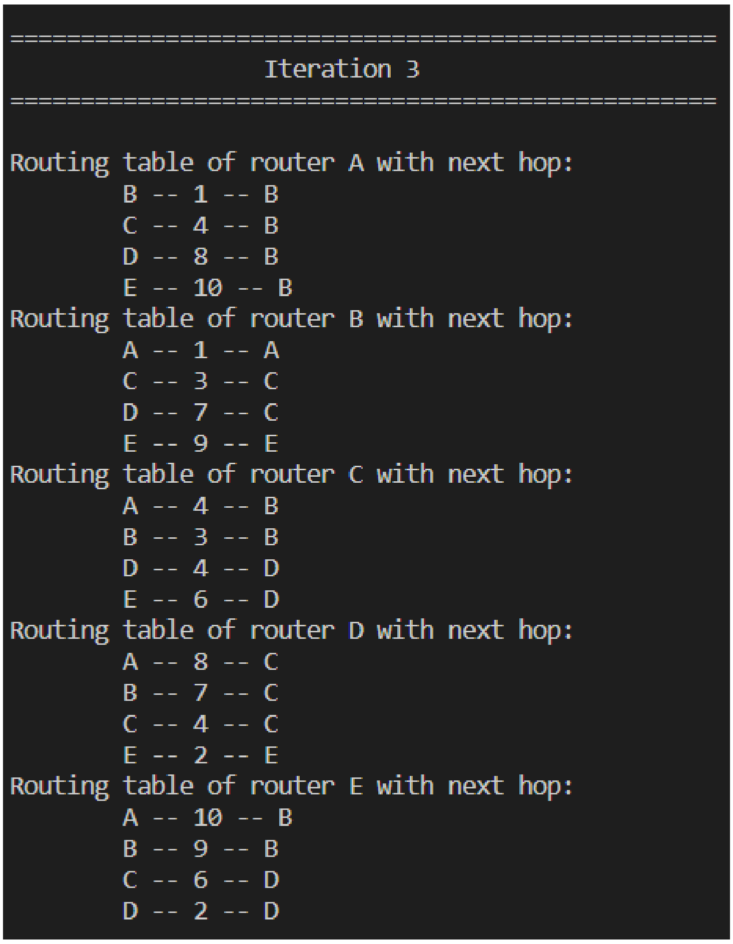
```
Initialised input:

Routing table of router A:
B -- 1
C -- 5
D -- inf
E -- inf
Routing table of router B:
A -- 1
C -- 3
D -- inf
E -- 9
Routing table of router C:
A -- 5
B -- 3
D -- 4
E -- inf
Routing table of router D:
A -- inf
B -- inf
C -- 4
E -- 2
Routing table of router E:
A -- inf
B -- 9
C -- inf
D -- 2
enter No of iterations: 4
```



```
==================================================
                  Iteration 1
==================================================

Routing table of router A with next hop:
        B -- 1 -- B
  *     C -- 4 -- B
  *     D -- 9 -- C
  *     E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
  *     D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
  *     A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
  *     E -- 6 -- D
Routing table of router D with next hop:
  *     A -- 9 -- C
  *     B -- 7 -- C
        C -- 4 -- C
        E -- 2 -- E
Routing table of router E with next hop:
  *     A -- 10 -- B
        B -- 9 -- B
  *     C -- 6 -- D
        D -- 2 -- D
```

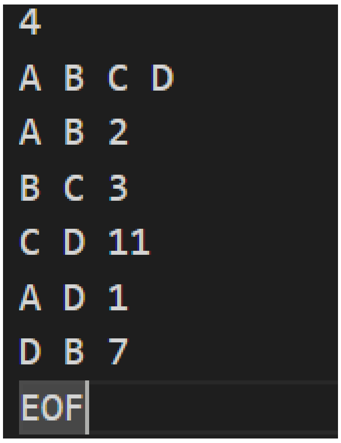


```
================================================
                  Iteration 2
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 4 -- B
  *     D -- 8 -- B
        E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
        D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
        A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
        E -- 6 -- D
Routing table of router D with next hop:
  *     A -- 8 -- C
        B -- 7 -- C
        C -- 4 -- C
        E -- 2 -- E
Routing table of router E with next hop:
        A -- 10 -- B
        B -- 9 -- B
        C -- 6 -- D
        D -- 2 -- D
```



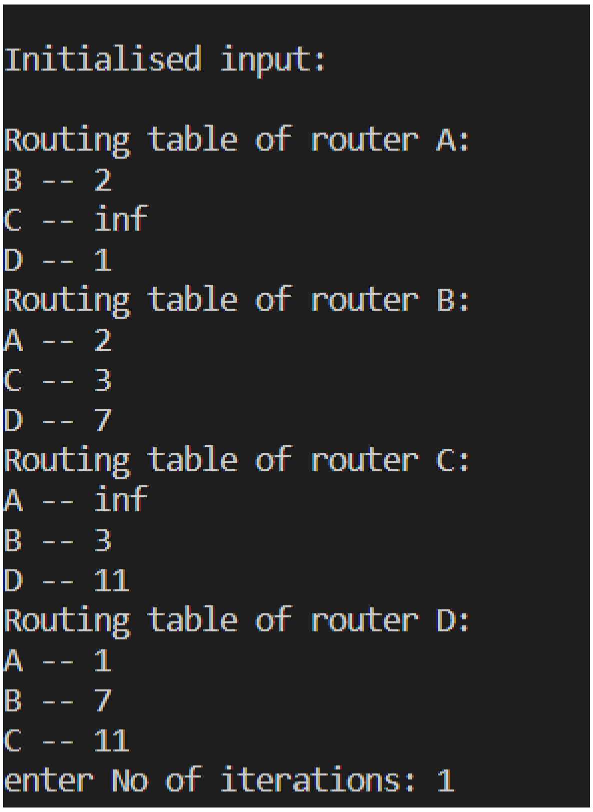
```
==================================================
                  Iteration 3
==================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 4 -- B
        D -- 8 -- B
        E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
        D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
        A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
        E -- 6 -- D
Routing table of router D with next hop:
        A -- 8 -- C
        B -- 7 -- C
        C -- 4 -- C
        E -- 2 -- E
Routing table of router E with next hop:
        A -- 10 -- B
        B -- 9 -- B
        C -- 6 -- D
        D -- 2 -- D
```

Input2.txt

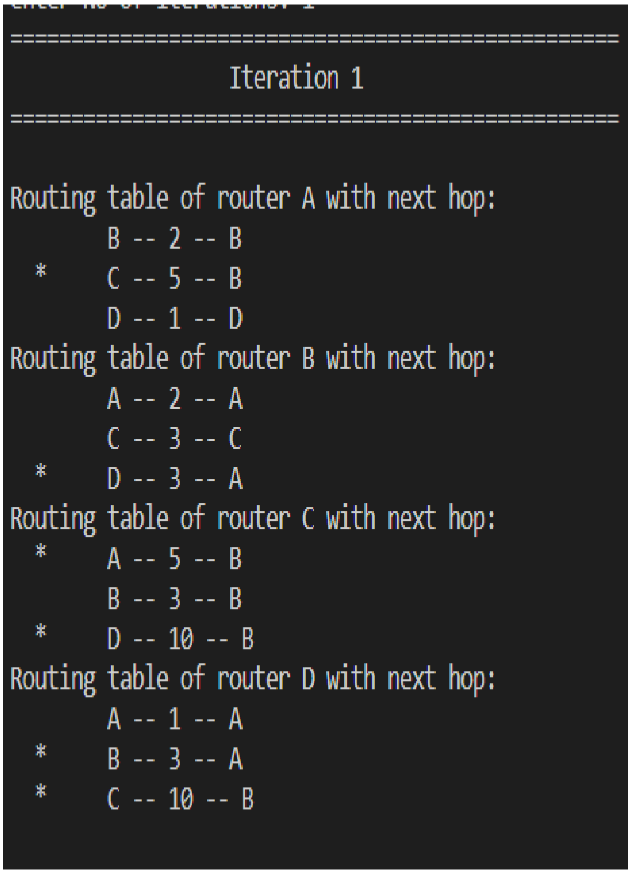


```
4
A B C D
A B 2
B C 3
C D 11
A D 1
D B 7
EOF
```

Output

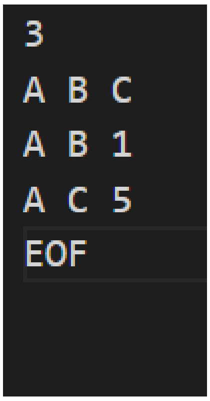


```
Initialised input:

Routing table of router A:
B -- 2
C -- inf
D -- 1
Routing table of router B:
A -- 2
C -- 3
D -- 7
Routing table of router C:
A -- inf
B -- 3
D -- 11
Routing table of router D:
A -- 1
B -- 7
C -- 11
enter No of iterations: 1
```



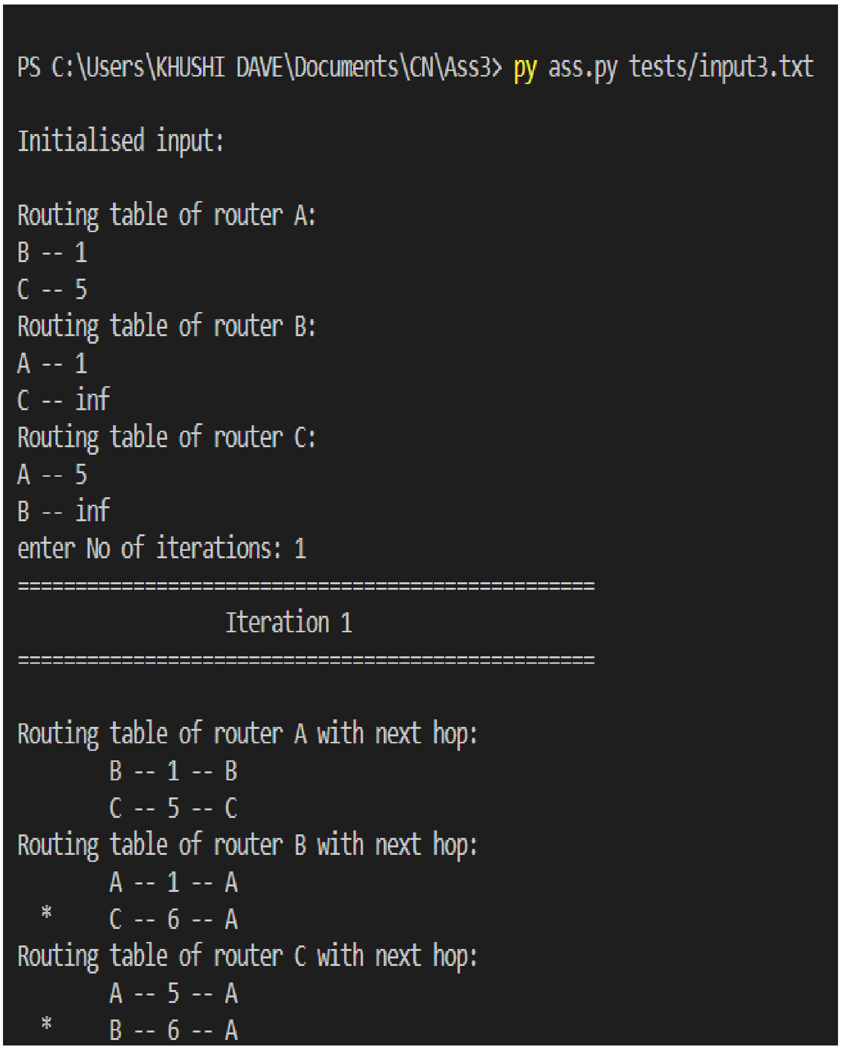
```
Enter No of iterations: 1
==================================================
                 Iteration 1
==================================================

Routing table of router A with next hop:
        B -- 2 -- B
  *     C -- 5 -- B
        D -- 1 -- D
Routing table of router B with next hop:
        A -- 2 -- A
        C -- 3 -- C
  *     D -- 3 -- A
Routing table of router C with next hop:
  *     A -- 5 -- B
        B -- 3 -- B
  *     D -- 10 -- B
Routing table of router D with next hop:
        A -- 1 -- A
  *     B -- 3 -- A
  *     C -- 10 -- B
```

Input3.txt



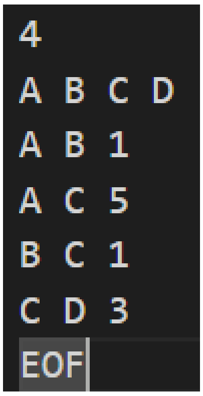
```
3
A B C
A B 1
A C 5
EOF
```

Output



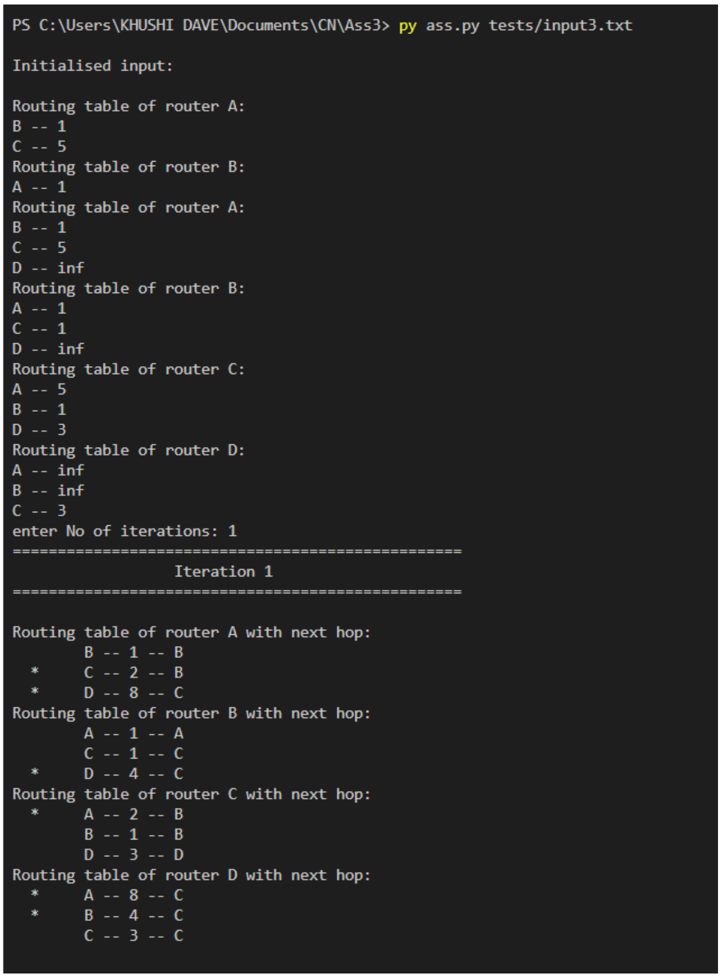
```
PS C:\Users\KHUSHI DAVE\Documents\CN\Ass3> py ass.py tests/input3.txt

Initialised input:

Routing table of router A:
B -- 1
C -- 5
Routing table of router B:
A -- 1
C -- inf
Routing table of router C:
A -- 5
B -- inf
enter No of iterations: 1
==================================================
         Iteration 1
==================================================

Routing table of router A with next hop:
    B -- 1 -- B
    C -- 5 -- C
Routing table of router B with next hop:
    A -- 1 -- A
 *  C -- 6 -- A
Routing table of router C with next hop:
    A -- 5 -- A
 *  B -- 6 -- A
```

Input4.txt



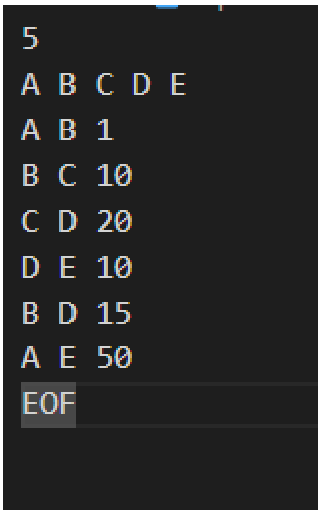
```
4
A B C D
A B 1
A C 5
B C 1
C D 3
EOF
```

Output

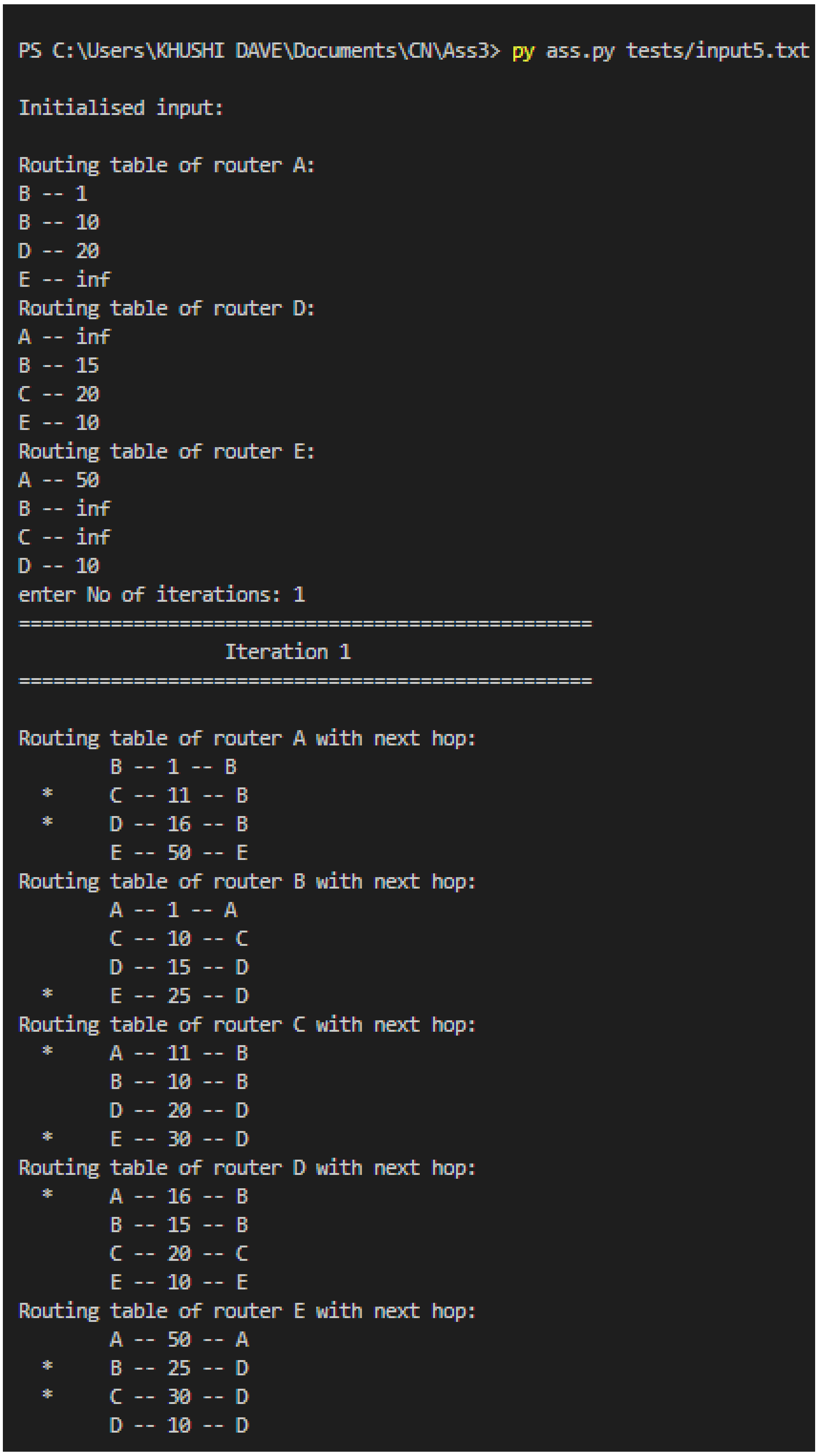


```
PS C:\Users\KHUSHI DAVE\Documents\CN\Ass3> py ass.py tests/input3.txt

Initialised input:

Routing table of router A:
B -- 1
C -- 5
Routing table of router B:
A -- 1
Routing table of router A:
B -- 1
C -- 5
D -- inf
Routing table of router B:
A -- 1
C -- 1
D -- inf
Routing table of router C:
A -- 5
B -- 1
D -- 3
Routing table of router D:
A -- inf
B -- inf
C -- 3
enter No of iterations: 1
==================================================
                  Iteration 1
==================================================

Routing table of router A with next hop:
        B -- 1 -- B
  *     C -- 2 -- B
  *     D -- 8 -- C
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 1 -- C
  *     D -- 4 -- C
Routing table of router C with next hop:
  *     A -- 2 -- B
        B -- 1 -- B
        D -- 3 -- D
Routing table of router D with next hop:
  *     A -- 8 -- C
  *     B -- 4 -- C
        C -- 3 -- C
```

Input5.txt



```
5
A B C D E
A B 1
B C 10
C D 20
D E 10
B D 15
A E 50
EOF
```



```
PS C:\Users\KHUSHI DAVE\Documents\CN\Ass3> py ass.py tests/input5.txt

Initialised input:

Routing table of router A:
B -- 1
B -- 10
D -- 20
E -- inf
Routing table of router D:
A -- inf
B -- 15
C -- 20
E -- 10
Routing table of router E:
A -- 50
B -- inf
C -- inf
D -- 10
enter No of iterations: 1
==================================================
                  Iteration 1
==================================================

Routing table of router A with next hop:
        B -- 1 -- B
  *     C -- 11 -- B
  *     D -- 16 -- B
        E -- 50 -- E
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 10 -- C
        D -- 15 -- D
  *     E -- 25 -- D
Routing table of router C with next hop:
  *     A -- 11 -- B
        B -- 10 -- B
        D -- 20 -- D
  *     E -- 30 -- D
Routing table of router D with next hop:
  *     A -- 16 -- B
        B -- 15 -- B
        C -- 20 -- C
        E -- 10 -- E
Routing table of router E with next hop:
        A -- 50 -- A
  *     B -- 25 -- D
  *     C -- 30 -- D
        D -- 10 -- D
```