**Distance Vector Routing Protocol**

Routers using distance-vector protocol do not have knowledge of the entire path to a destination. Instead they use two methods: Direction in which router or exit interface a packet should be forwarded, Distance from its destination.

Distance-vector protocols are based on calculating the direction and distance to any link in a network. "Direction" usually means the next hop address and the exit interface. "Distance" is a measure of the cost to reach a certain node. The least cost route between any two nodes is the route with minimum distance. Each node maintains a vector (table) of minimum distance to every node. The cost of reaching a destination is calculated using various route metrics. RIP uses the hop count of the destination whereas IGRP considers other information such as node delay and available bandwidth.

Updates are performed periodically in a distance-vector protocol where all or part of a router's routing table is sent to all its neighbors that are configured to use the same distance-vector routing protocol. Once a router has this information it can amend its own routing table to reflect the changes and then inform its neighbors of the changes. This process has been described as ‘routing by rumor’ because routers are relying on the information they receive from other routers and cannot determine if the information is valid and true. There are several features which can be used to help with instability and inaccurate routing information.

A simple routing protocol that uses distance or hop count as its primary metric for determining the best forwarding path. RIP, IGRP and EIGRP are examples. A distance vector protocol routinely sends its neighboring routers copies of its routing tables to keep them up-to-date.

**RIP:**

The Routing Information Protocol (RIP) is one of the oldest distance-vector routing protocols which employ the hop count as a routing metric. RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination. The maximum number of hops allowed for RIP is 15, which limits the size of networks that RIP can support. A hop count of 16 is considered an infinite distance and the route is considered unreachable. RIP implements the split horizon, route poisoning and hold down mechanisms to prevent incorrect routing information from being propagated. This is the process, which typically makes a request for information. After getting the response, this process may terminate or may do some other processing.

**Step-1:**

Install python

Set environment paths

pip install apscheduler==2.1.2

**Execution**

Create a data file for each node as suggested in problem statement (node\_name.dat).

Open a new terminal for each node i.e 6

Run the python code with address and data file

e.g. python dv.py localhost a.dat

Run the above script for every node (.dat file) in 6 command prompts

**Note**: The infinity is set with value 9999

**DATA FILES**

Here we are setting the data files with the below data according to the figure given:

**a.dat**



**b.dat**



**c.dat**



**d.dat**



**e.dat**

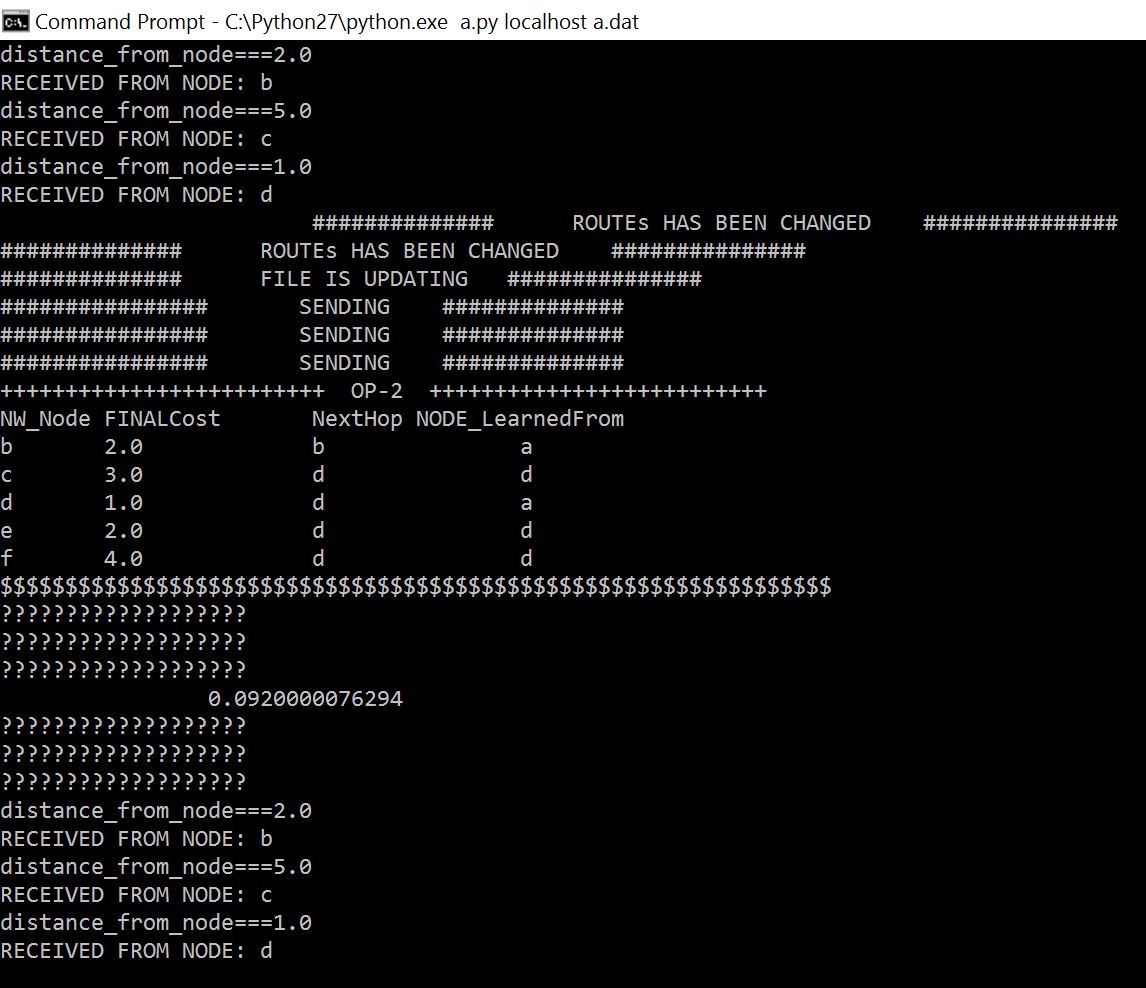


**f.dat**

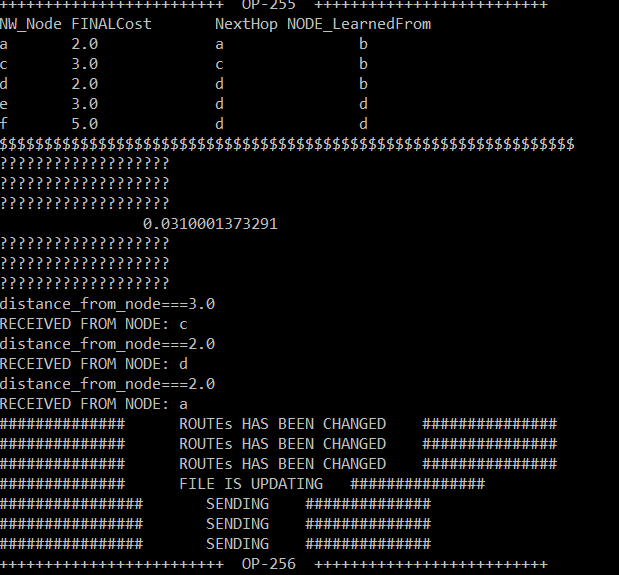


**Output Screenshots for above data files:**

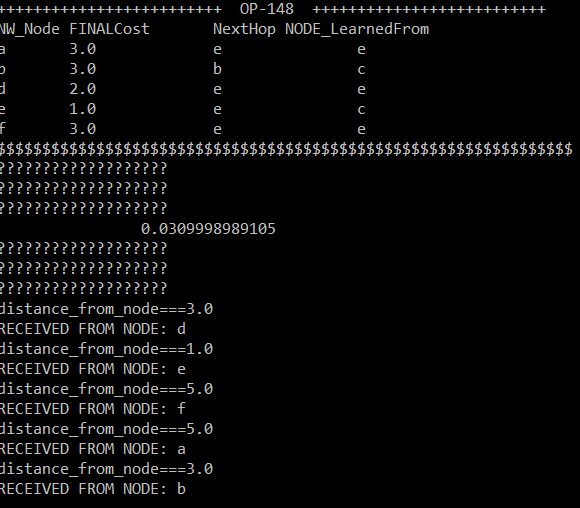
**For a.dat data file : Node A**

****

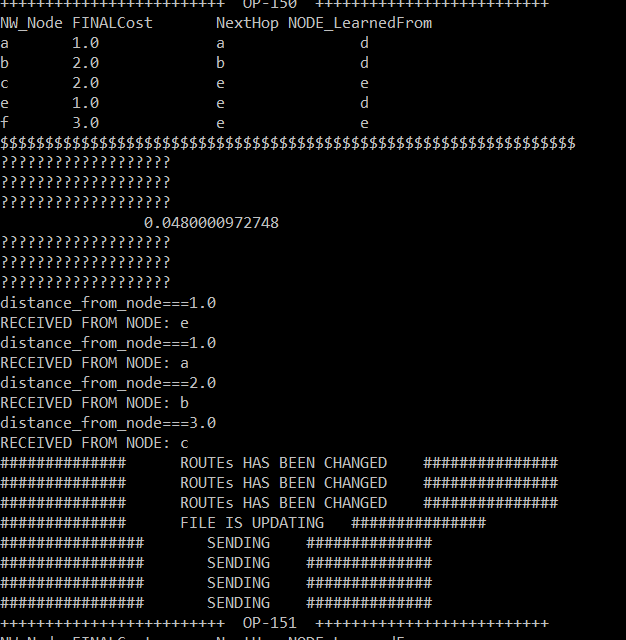
**For b.dat data file : Node B**

****

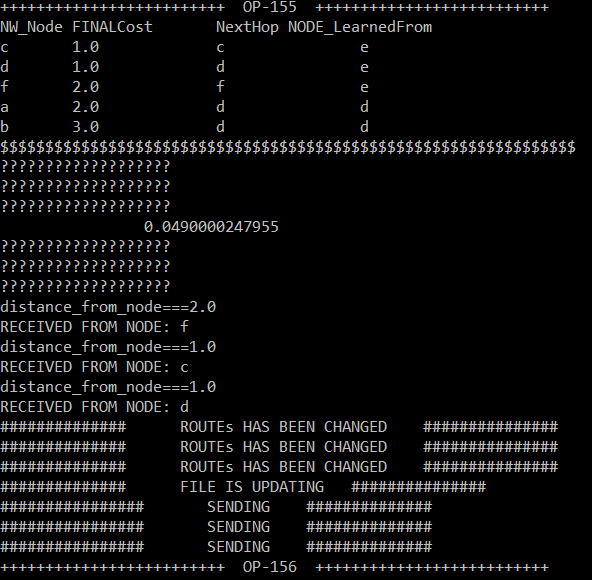
**For c.dat data file : Node C**

****

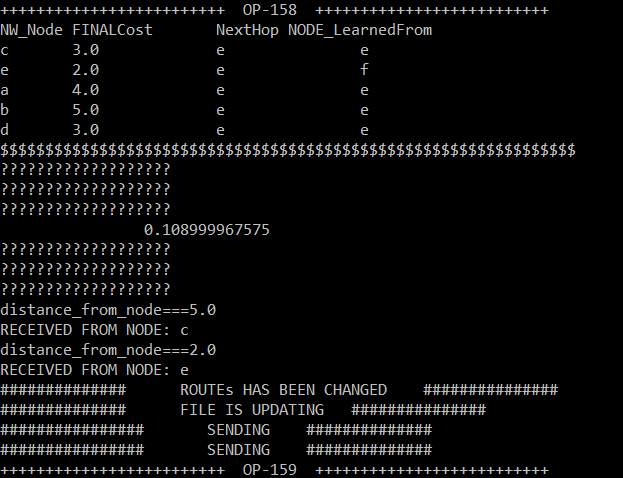
**For d.dat data file : Node D**

****

**For e.dat data file : Node E**

****

**For f.dat data file : Node F**

****

**--------------------------------**

**2.DATA FILES**

Here we are setting the data files with the below data according to the figure given:

**CHANGING COST A-D from 1 to 7**

**a.dat**

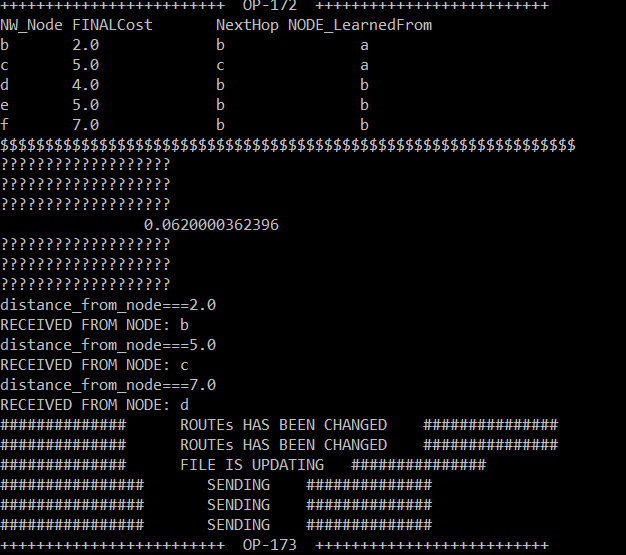


**d.dat**

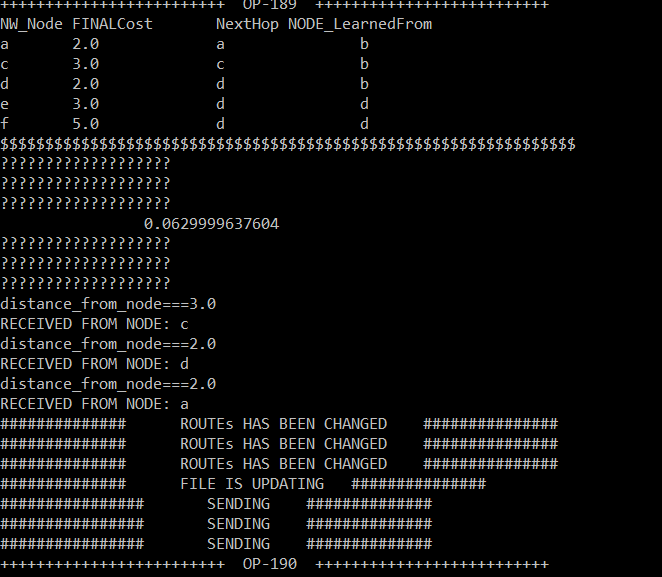


**Output Screenshots for above data files:**

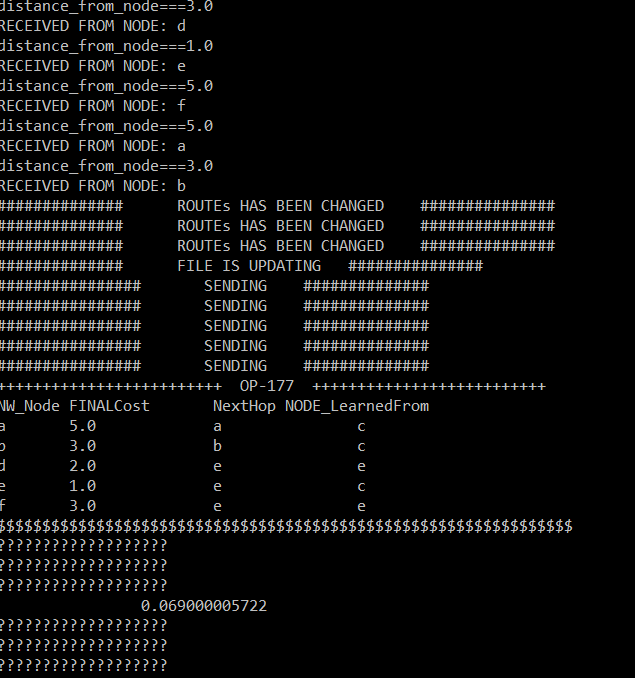
**For a.dat data file : Node A**

****

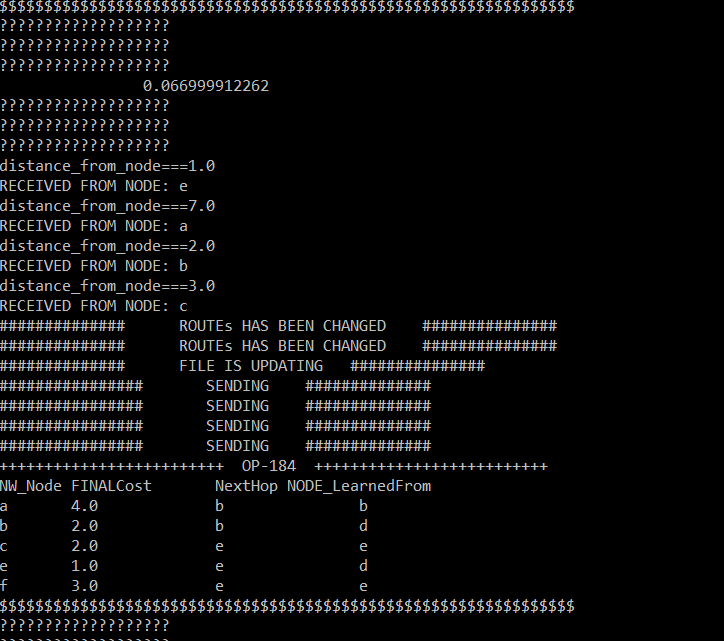
**For b.dat data file : Node B**

****

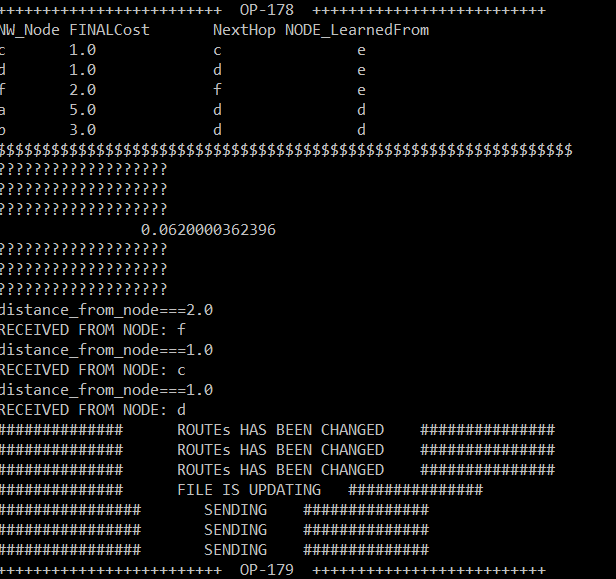
**For c.dat data file : Node C**

****

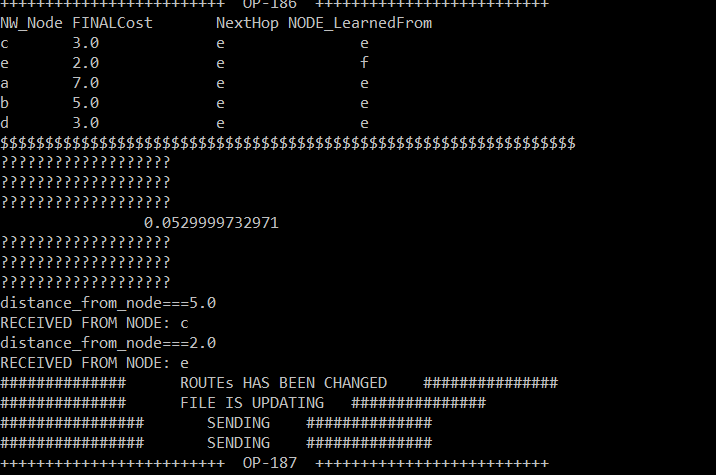
**For d.dat data file : Node D**

****

**For e.dat data file : Node E**

****

**For f.dat data file : Node F**

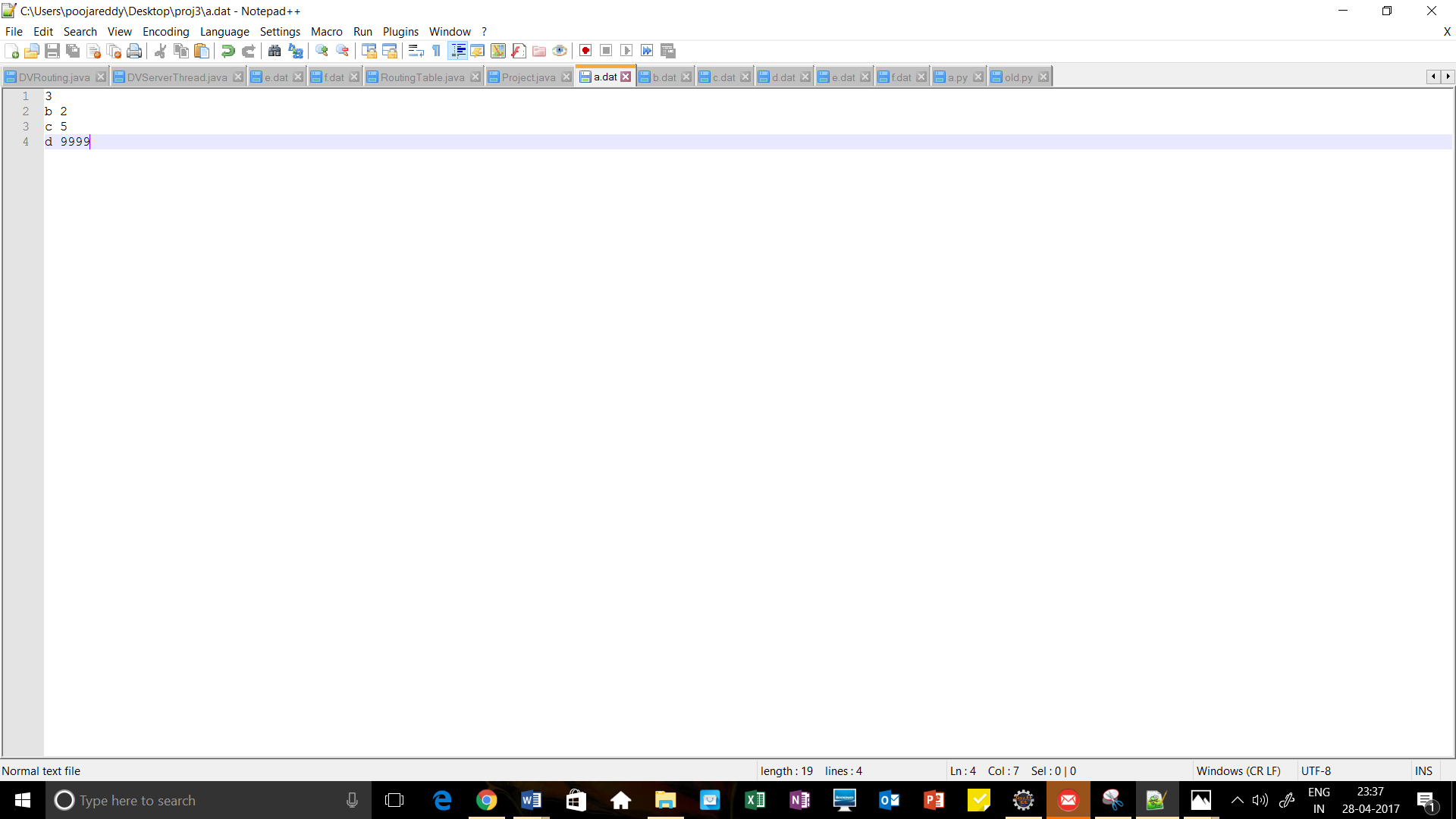
****

**-------------------------------------------------------------------------------------------------------------------------------------------**

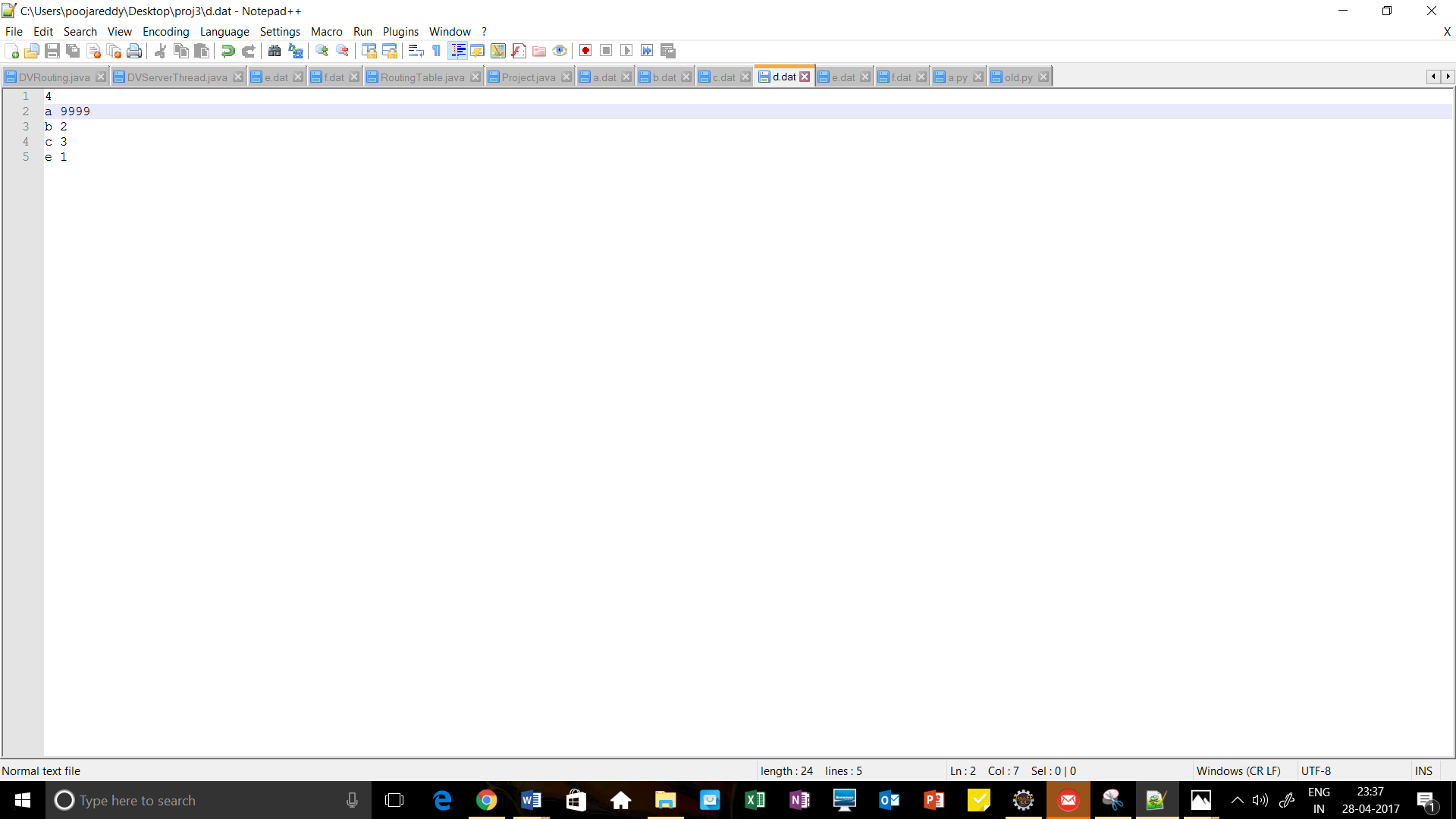
**3.Data files**

**Here the .dat files is set according to the figures below:**

**a.dat**

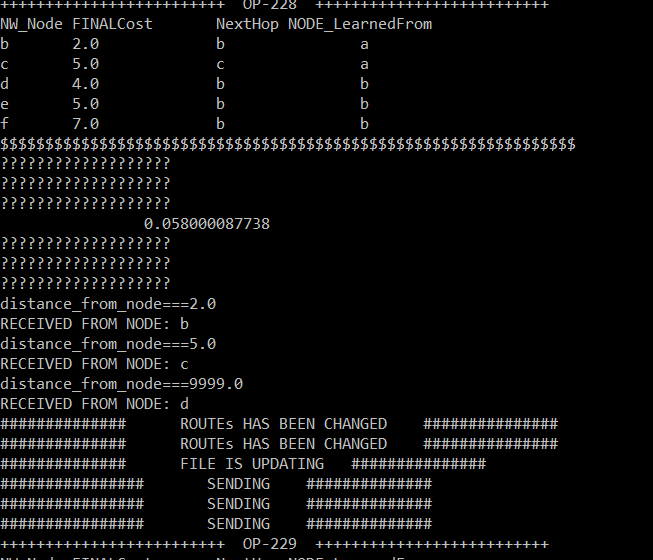
****

**d.dat**

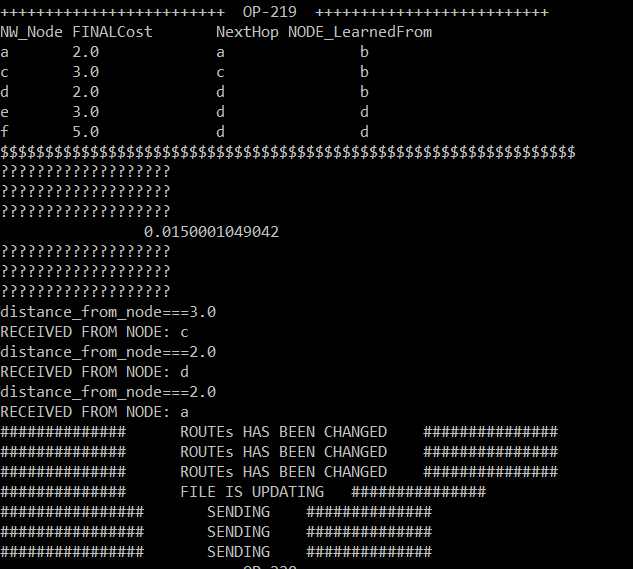
****

**Output Screenshots for above data files:**

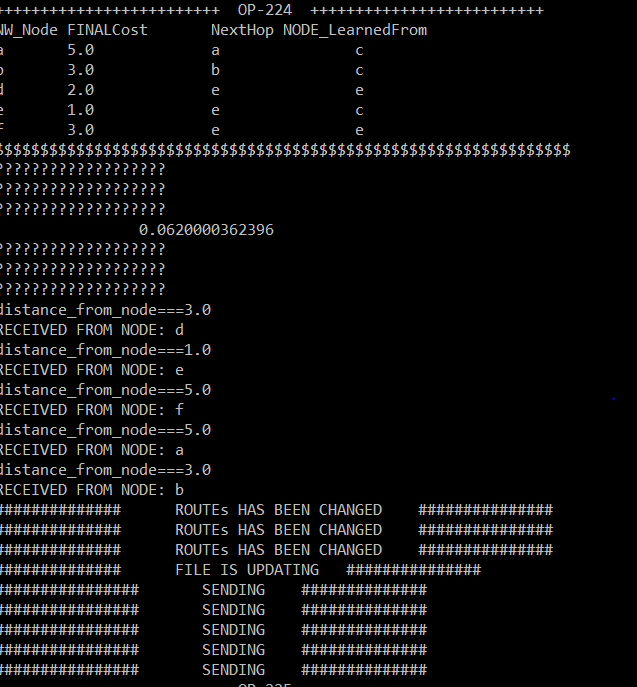
**For a.dat data file : Node A**

****

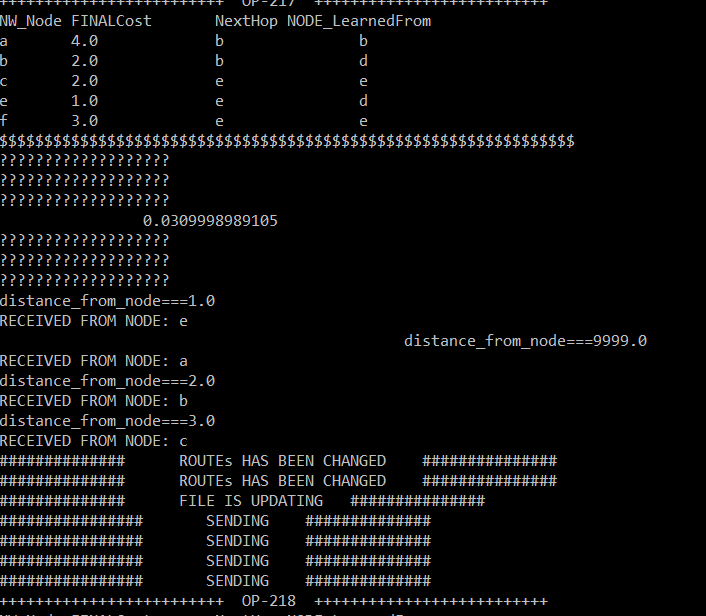
**For b.dat data file : Node B**

****

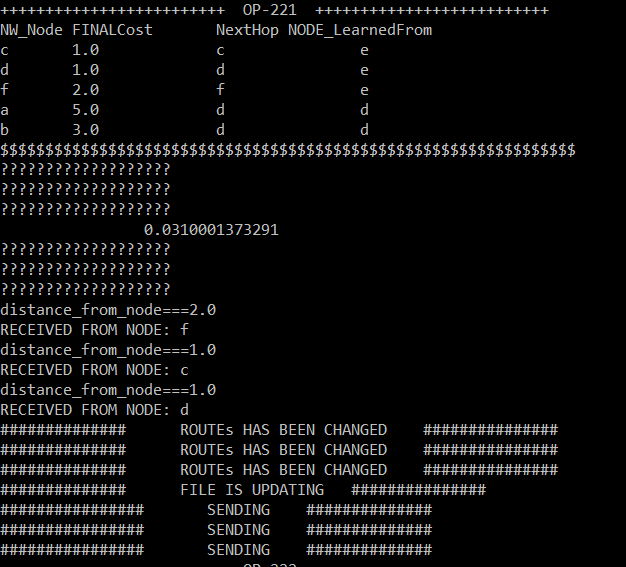
**For c.dat data file : Node C**

****

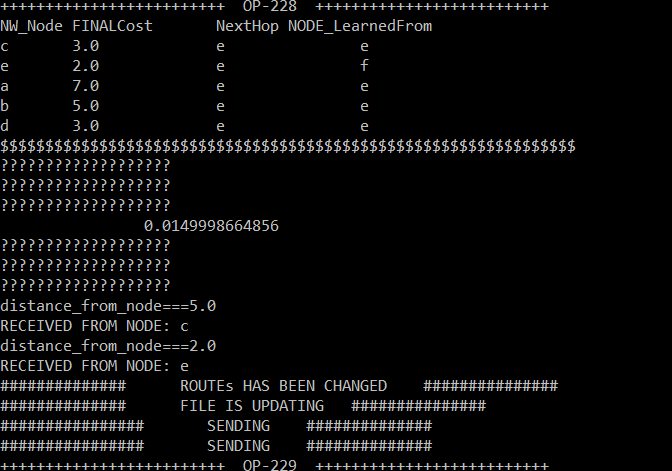
**For d.dat data file : Node D**

****

**For e.dat data file : Node E**

****

**For f.dat data file : Node F**

****