C1)

a) Included in the folder as server.py and client.py

b) The time taken for finding the shortest path is ~8sec. I have computed it by finding out the created time and then the last modified time and then computed the difference between them.

   Use the commands la -I to find the inode number and then use debugfs -R ‘stat <inode number>’ /dev/sda1

c) H1:

```
r4,13,
r1,2,172.1.0.2
r2,12,
r3,8,
h2,15,
h1,0,172.1.0.1
```

R1:

```
r4,11,
r1,0,172.1.0.2
r2,10,173.1.0.2
r3,6,174.1.0.2
h2,13,
h1,2,172.1.0.1
```

R2:

```
r4,4,177.1.0.1
r1,10,172.1.0.2
r2,0,173.1.0.2
r3,9,
h2,6,
h1,12,
```

R3:

```
r4,5,177.1.0.1
r1,6,172.1.0.2
r2,9,
r3,0,174.1.0.2
h2,7,
h1,8,
```

R4:

```
r4,0,177.1.0.1

r1,11,

r2,4,173.1.0.2
r3,5,174.1.0.2
h2,2,177.1.0.2
h1,13,
```

H2:

```
r4,2,177.1.0.1
r1,13,
r2,6,
r3,7,
h2,0,177.1.0.2
h1,15,
```

C2)

a) The convergence time was ~15sec

b) H1:

```
r4,8,
r1,2,172.1.0.2
r2,12,
r3,3,
h2,10,
        h1,0,172.1.0.1

R1:

        r4,6,
        r1,0,172.1.0.2
        r2,10,173.1.0.2
        r3,1,174.1.0.2
        h2,8,
        h1,2,172.1.0.1

R2:

        r4,4,177.1.0.1
        r1,10,172.1.0.2
        r2,0,173.1.0.2
        r3,9,
        h2,6,
        h1,12,

R3:
        r4,5,177.1.0.1
        r1,1,172.1.0.2
        r2,9,
        r3,0,174.1.0.2
        h2,7,
        h1,3,
R4:
        r4,0,177.1.0.1
        r1,6,
        r2,4,173.1.0.2
        r3,5,174.1.0.2
        h2,2,177.1.0.2
        h1,8,
H2:
        r4,2,177.1.0.1
        r1,8,
        r2,6,
        r3,7,
        h2,0,177.1.0.2
        h1,10,
```

C3)

It is quite possible that if the weights are negative then there is a negative cycle also possible. Bellman ford algorithm can calculate the shortest path even if the weights are negative, but if these negative weights form a cycle then it would be a problem. In such cases I would detect if a negative cycle exits in the network every time I compute the algorithm and if the negative cycle exits , I would terminate the algorithm at that point itself. This is the way I would handle negative edges.

In order to run the program follow the below steps:

1) Navigate to the partc folder
2) Sudo python start.py
3) Mininet>source enable_forward.sh
4) Mininet> source interface.sh
5) Mininet> source routes.sh
6) Mininet>source nat.sh
7) Mininet>pingall --- to check if all the nodes are pinging
8) Mininet>source cmd.sh
9) Now open another terminal and check for the updated tables in the corresponding files with names *.txt

References :

https://www.geeksforgeeks.org/socket-programming-python/