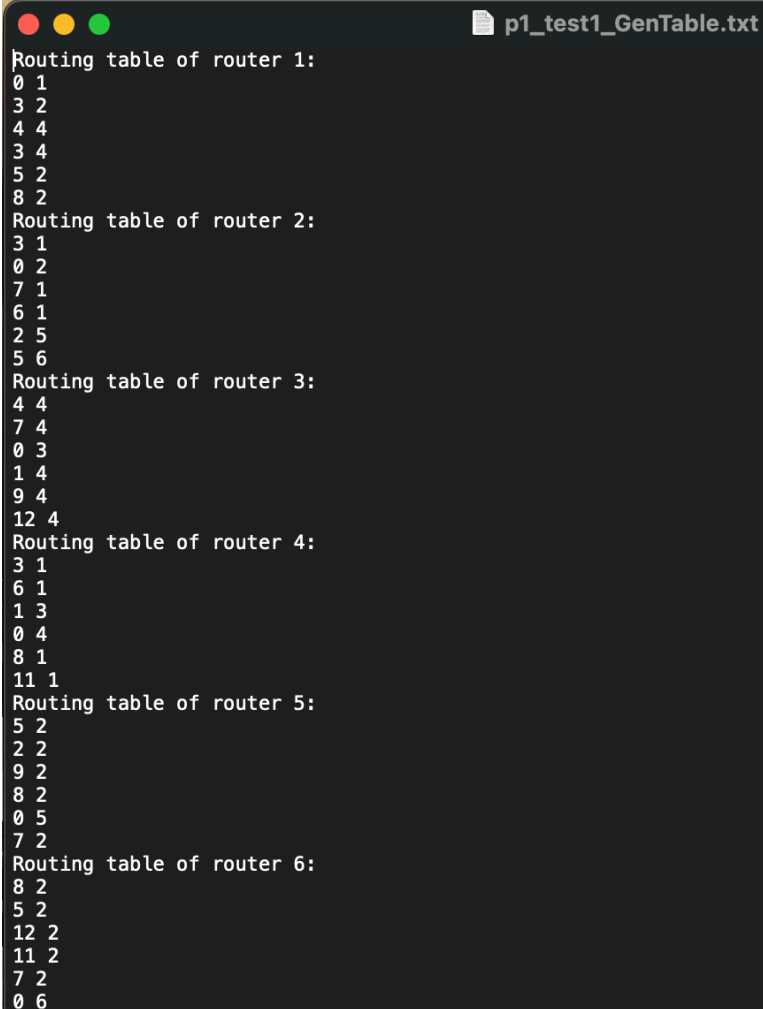1.

(1) Import sys and use sys.argv[1] to read the input file name from terminal.
(2) Read file and do some preprocessing to get total number or routers as N, and save the distance matrix as a 2D list of int.
(3) Init final distance and next node 2D list to save results.
(4) Run the processes below (5) ~ (8) with every nodes as the source.
(5) Initialize temp arrays of distance with sys.maxsize which is a very large number, set the distance of source to 0. Another array called nextHop to save the next node number, initialized with the initial first hop of every destination. Finally, there is an array done with all false values to mark whether a node has finished discovering.
(6) Loop to update all minimum distance by: Finding min distance node. => Setting the minimum distance node as true in done. => Update distance and next hop value of the adjacent nodes when the distance gets smaller.
(7) Correct the index of next hop, which starts from 1 by adding 1.
(8) Save the distance and next hop value result to the array created in (3).
(9) Write the result to the correct file path "p1_test*_GenTable.txt" in the same format as the golden example.

Result: (test 1 for example)

```
● ● ●                    📄 p1_test1_GenTable.txt
Routing table of router 1:
0 1
3 2
4 4
3 4
5 2
8 2
Routing table of router 2:
3 1
0 2
7 1
6 1
2 5
5 6
Routing table of router 3:
4 4
7 4
0 3
1 4
9 4
12 4
Routing table of router 4:
3 1
6 1
1 3
0 4
8 1
11 1
Routing table of router 5:
5 2
2 2
9 2
8 2
0 5
7 2
Routing table of router 6:
8 2
5 2
12 2
11 2
7 2
0 6
```

2.

(1) Import sys and use sys.argv[1] to read the input file name, sys.argv[2] to read the router that failed from terminal and save it as rmv.
(2) Read file and do some preprocessing to get total number or routers as N, and save the distance matrix as a 2D list of int.
(3) If rmv router exist, set router distance in distance matrix that are related to rmv router to −1 to disable the failed router.
(4) Run the same process in 1. (3) ~ (8).
(5) If values in nextHop_mat equals 0, change it to −1. If values in distance_mat equals the original large number set in (5), change it to −1. Which means one router be not able to reach another routers.
(6) Write the result to the correct file path "p1_test*_RmRouter(rmv).txt" in the same format as the golden example.

Result: (test 1 remove 2 for example)

```
● ● ●                          📄 p1_test1_RmRouter2.txt
Routing table of router 1:
0 1
-1 -1
4 4
3 4
-1 -1
-1 -1
Routing table of router 3:
4 4
-1 -1
0 3
1 4
-1 -1
-1 -1
Routing table of router 4:
3 1
-1 -1
1 3
0 4
-1 -1
-1 -1
Routing table of router 5:
-1 -1
-1 -1
-1 -1
-1 -1
0 5
-1 -1
Routing table of router 6:
-1 -1
-1 -1
-1 -1
-1 -1
-1 -1
0 6
```