Lab 12
PF – BSDS

```
LENGTH = randint(5,9)
d = [[randint(1,9) for i in range(LENGTH)] for j in range(LENGTH)]
for i in range(LENGTH):
      d[i][i] = 0
```

Consider above code for next tasks. Consider d is distance between cities. There is one direct path from every city to every other city and there is one indirect path that is thorough any other city.

Task 1:

a. Write a function with three parameters distance, city 1 and city 2. The distance is the 2D list (given), city 1 and city 2 is some number from 0 to LENGTH - 1. Print all distances from city 1 to city 2. One distance will be direct. Remaining distances will be one indirect. For example, there will be one direct distance between city 2 and city 4; whereas, there will be a distance between city 2 and city 4 via city 2 to city 6 and city 6 to city 2 and the total distance will be the sum of two distances. The output should be like:

Sample Run:

```
0 1 9 6 6 2 4 8
7 0 9 7 5 5 2 3
8 6 0 3 1 9 3 4
7 5 9 0 6 8 5 2
5 5 3 6 0 3 7 2
4 9 2 4 7 0 2 4
9 7 9 5 5 3 0 7
9 5 6 5 9 8 4 0
Distance between City 2 and City 4
Direct Distance: 1
Indirect Distances
Via City 0: 14
Via City 1: 11
Via City 3: 9
Via City 5: 16
Via City 6: 8
Via City 7: 13
```

b. Modify function made in task 1 (a). Print shortest distance between city A and city B. Also print whether distance is direct or indirect. In case, of indirect shortest distance print the city number to be used between city A and city B.

Sample Runs:

```
0 6 4 5 3 2 6 9
6 0 6 2 3 9 3 8
2 4 0 1 7 3 1 4
1 9 5 0 7 3 7 2
5 6 5 6 0 1 6 1
1 6 2 2 6 0 8 7
9 2 6 1 9 9 0 1
5 7 9 8 4 7 4 0
Shortest distance between city 0 and city 1 is 6.
Shortest distance between city 0 and city 2 is 4.
Shortest distance between city 0 and city 3 is 4 via city 5.
Shortest distance between city 0 and city 4 is 3.
Shortest distance between city 0 and city 5 is 2.
Shortest distance between city 0 and city 6 is 5 via city 2.
Shortest distance between city 0 and city 7 is 4 via city 4.
```

c. Print the name of two cities having shortest direct distance

Sample Run:

```
0 6 7 3 3 6 3 3
6 0 3 7 2 6 4 8
1 5 0 7 4 2 6 7
7 8 9 0 1 7 3 3
7 7 5 5 0 8 4 1
5 9 8 2 9 0 7 7
9 4 1 7 4 8 0 8
5 4 9 8 9 1 6 0
City 2 and City 0 has shortest direct distance 1
```

For next task, place -1 at random showing there is no direct path between two cities. For example,

```
0  4 -1  6
5  0  7 -1
-1 3  0 -1
...
```

City 0 has direct path with city 1 and city 3 but no direct path with city 2.
Similarly, city 1 has direct path with cityd 0 and city 2 but no direct path with city 3
Similarly, city 2 has direct path with city 1 but no direct path between city 0 and city 3.

Task 2:

a. Write code to print cities having direct link.

Sample Run:

```
0 9 4 4 -1 9 1 -1
1 0 8 6 7 8 3 6
9 3 0 3 2 5 -1 -1
3 3 6 0 8 -1 6 3
-1 2 5 5 0 1 8 1
6 3 3 1 5 0 5 9
6 4 -1 9 -1 9 0 3
7 1 2 -1 8 7 7 0
City 0 has direct link with 1 2 3 5 6
City 1 has direct link with 0 2 3 4 5 6 7
City 2 has direct link with 0 1 3 4 5
City 3 has direct link with 0 1 2 4 6 7
City 4 has direct link with 1 2 3 5 6 7
City 5 has direct link with 0 1 2 3 4 6 7
City 6 has direct link with 0 1 3 5 7
City 7 has direct link with 0 1 2 4 5 6
```

b. Write code to print cities having no direct link but one indirect link.

Sample Run:

```
0 9 -1 5 5 -1 2
5 0 -1 -1 -1 5 9
-1 5 0 6 -1 -1 6
1 5 2 0 -1 4 4
9 -1 9 3 0 4 -1
6 6 -1 9 6 0 5
1 9 6 8 7 -1 0
City 0 has indirect link with City 2 via 1
City 0 has indirect link with City 5 via 1
City 1 has indirect link with City 2 via 0
City 1 has indirect link with City 3 via 0
City 1 has indirect link with City 4 via 0
City 2 has indirect link with City 0 via 3
City 2 has indirect link with City 4 via 3
City 2 has indirect link with City 5 via 3
City 3 has indirect link with City 4 via 0
City 4 has indirect link with City 1 via 0
```

```
City 4 has indirect link with City 6 via 0
City 5 has indirect link with City 2 via 1
City 6 has indirect link with City 5 via 0
```