Session 4: Lists and Dictionaries

1. Syntax for Lists

```
[1]: li=['China','Italy','Spain']

[2]: li[0]

'China'

[3]: li[1]

'Italy'

[4]: len(li)

3

[5]: li.append('Iran')

[6]: li

['China', 'Italy', 'Spain', 'Iran']

[7]: li[1:3]

['Italy', 'Spain']

[8]: li[-1]

'Iran'

[9]: li[1:-1]

['Italy', 'Spain']

[10]: 'US' in li

False

[11]: 'US' not in li

True
```

Q1. Basic Database of Epidemic Cases in the US

Working with your discussion group, create the backbone of a data management system for a new epidemic in the US. Concretely speaking, you should write the following three functions.

a) A function called `addCase` with two inputs:

- name: the name of the patient.
- state: the state the person is from.

The function should add this entry to the database.

b) A function called `retrieveCase` with one input:

- index: a number associated with the case. The first case is given index 0, the second case index 1, etc.

The function should return a list of two elements describing the case associated with this index. The first element is the patient's name and the second is the State.

c) A function called `totalCases` with no input. The function returns an integer representing the total number of cases so far.

See the test code below for sample outputs.

```
[13]: addCase('Alice','New York')
      addCase('Bob','California')
      addCase('Jack','New York')
      totalCases()

3

[14]: retrieveCase(0)

Case 0 is Alice in New York.


[15]: retrieveCase(1)

Case 1 is Bob in California.


[16]: retrieveCase(2)

Case 2 is Jack in New York.


[17]: retrieveCase(3)

Invalid index. Need to be between 0 and 2.
```

2. Syntax for Dictionaries

```
[18]: d={'China':80000,'Italy':50000,'Spain':20000}

[19]: d['China']

80000

[20]: len(d)

3

[21]: d['Iran']=20000

[22]: d

{'China': 80000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[23]: d['China']=81000

[24]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[25]: 'USA' in d

False

[26]: 'USA' not in d

True

[27]: d.keys()

dict_keys(['China', 'Italy', 'Spain', 'Iran'])

[28]: list(d.keys())

['China', 'Italy', 'Spain', 'Iran']

[29]: d.values()

dict_values([81000, 50000, 20000, 20000])

[30]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[31]: d[5]=['A',3000]

[32]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000, 5: ['A', 3000]}
```

Q2. Adding Capability to Query by State

Working with your discussion group, add the following two functions to the data management system.

a) A function called `totalInState` with one input, state. The function should return the total number of cases in this State so far.

b) A function called `patientsInState` with one input, state. The function should return a list of names corresponding to all the cases in this State.

You should also copy the three other functions from Q1 into your code cell and modify them appropriately, so as to also keep track of cases by State as new cases are being added. See the following testing code for sample outputs.

```
[34]: addCase('Alice','New York')
      addCase('Bob','California')
      addCase('Jack','New York')
      totalCases()

3

[35]: totalInState('Michigan')

0

[36]: totalInState('New York')

2

[37]: patientsInState('New York')

['Alice', 'Jack']
```

(Optional) Q3. A Simple User Interface

Create a simple user interface to the data management system from Q1 and Q2, by writing a function called `caseManagement` with no input arguments. The function should call the appropriate function based on user's inputs. See the following testing code for examples of how the function behaves. Write your function in a code cell, which should also initialize the database to be empty.

```
[39]: caseManagement()

Enter command (add, retrieve, total, totalInState, patientsInState): add
Enter patient name: Alice
Enter State: New York


[40]: caseManagement()

Enter command (add, retrieve, total, totalInState, patientsInState): Charlie
Invalid command


[41]: caseManagement()

Enter command (add, retrieve, total, totalInState, patientsInState): add
Enter patient name: Charlie
Enter State: New York


[42]: caseManagement()

Enter command (add, retrieve, total, totalInState, patientsInState): patientsInState
Enter State: New York
List of patients in New York: ['Alice', 'Charlie']


[43]: caseManagement()

Enter command (add, retrieve, total, totalInState, patientsInState): total
Total number of cases: 2
```