**Airline Dictionary Practice Problems**

1) Write a function **def reportAllFlightsFromTo** **(flightsD,fromCity,toCity):** to print all of the flights specified in fromCity and toCity for all airlines in the following format:

>>> reportAllFlightsFromTo(flightsD,"PHX","MDW")

PHX MDW

NUM DEP ARR AIRLINE

4335 450 745 Southwestern

1096 900 1255 Delta

1226 950 1345 Delta

>>> reportAllFlightsFromTo(flightsD,"MDW","IND")

MDW IND

NUM DEP ARR AIRLINE

1009 1850 1955 Delta

9001 2145 2255 Delta

2) Write a function **def updateDaylightSavingsTimes(flightsD,AirportsNeedingDST,Increment):** that will update departure and arrival times by Increment for only the airports passed in the list AirportsNeedingDST. Note only airports included in the **AirportsNeedingDST** are actually updated in flightsD (e.g. MDW not updated)

>>> NeedDST=["IND","PHX","ATL","HOU","BNA","SAT","LAX","PHL"]

>>> flightsD

{'American': {**5577: [['ATL', 1100], ['HOU', 1350]]**, 1102: [['BNA', 1100], ['HOU', 1450]], 2100: [['BNA', 900], ['IND', 1115]], 7765: [['IND', 1850], ['CHA', 2105]], 4311: [['HOU', 905], ['ATL', 1255]], 3321: [['HOU', 1335], ['ATL', 1615]], 2133: [['BNA', 900], ['IND', 1115]]}, 'Southwestern': {2121: [['MDW', 430], ['SAT', 825]], **4335: [['PHX', 450], ['MDW', 745]],** 1102: [['MDW', 1100], ['PHX', 1450]], 1111: [['SAT', 430], ['MDW', 825]]}, 'Delta': {1776: [['PHL', 1350], ['RAP', 1610]], 1009: [['MDW', 1850], ['IND', 1955]], 1445: [['ATL', 1135], ['LAX', 1810]], 1096: [['PHX', 900], ['MDW', 1255]], 9001: [['MDW', 2145], ['IND', 2255]], 1226: [['PHX', 950], ['MDW', 1345]], 1885: [['ATL', 1305], ['LAX', 2000]], 1102: [['IND', 1850], ['MDW', 1955]]}}

>>> updateDaylightSavingsTimes(flightsD,NeedDST,100)

>>> flightsD

{'American': {**5577: [['ATL', 1200], ['HOU', 1450]],** 1102: [['BNA', 1200], ['HOU', 1550]], 2100: [['BNA', 1000], ['IND', 1215]], 7765: [['IND',1950], ['CHA', 2105]], 4311: [['HOU', 1005], ['ATL', 1355]], 3321: [['HOU',1435], ['ATL', 1715]], 2133: [['BNA', 1000], ['IND', 1215]]}, 'Southwestern': {2121: [['MDW', 430], ['SAT', 925]], **4335: [['PHX', 550], ['MDW', 745]],** 1102: [['MDW', 1100], ['PHX', 1550]], 1111: [['SAT', 530], ['MDW', 825]]}, 'Delta': {1776: [['PHL', 1450], ['RAP', 1610]], 1009: [['MDW', 1850], ['IND', 2055]], 1445: [['ATL', 1235], ['LAX', 1910]], 1096: [['PHX',1000], ['MDW', 1255]], 9001: [['MDW', 2145], ['IND', 2355]], 1226: [['PHX',1050], ['MDW', 1345]], 1885: [['ATL', 1405], ['LAX', 2100]], 1102: [['IND', 1950], ['MDW', 1955]]}}

3) Write a function **def createNewFlight(flightsD,airline,flight,depCity,depTime,arrCity,arrTime):** that adds new flight information to the dictionary-based data structure. (Could be written as a 1 liner!)

>>> flightsD

{'American': {5577: [['ATL', 1100], ['HOU', 1350]], 1102: [['BNA', 1100], ['HOU', 1450]], 2100: [['BNA', 900], ['IND', 1115]], 7765: [['IND', 1850], ['CHA', 2105]], 4311: [['HOU', 905], ['ATL', 1255]], 3321: [['HOU', 1335], ['ATL', 1615]], 2133: [['BNA', 900], ['IND', 1115]]}, 'Southwestern': {2121: [['MDW', 430], ['SAT', 825]], 4335: [['PHX', 450], ['MDW', 745]], 1102: [['MDW', 1100], ['PHX', 1450]], 1111: [['SAT', 430], ['MDW', 825]]}, 'Delta': {1776: [['PHL', 1350], ['RAP', 1610]], 1009: [['MDW', 1850], ['IND', 1955]], 1445: [['ATL', 1135], ['LAX', 1810]], 1096: [['PHX', 900], ['MDW', 1255]], 9001: [['MDW', 2145], ['IND', 2255]], 1226: [['PHX', 950], ['MDW', 1345]], 1885: [['ATL', 1305], ['LAX', 2000]], 1102: [['IND', 1850], ['MDW', 1955]]}}

>>> createNewFlight(flightsD,"JET BLUE",1010,"IND",605,"MDW",705)

>>> flightsD

{'American': {5577: [['ATL', 1100], ['HOU', 1350]], 1102: [['BNA', 1100], ['HOU', 1450]], 2100: [['BNA', 900], ['IND', 1115]], 7765: [['IND', 1850], ['CHA', 2105]], 4311: [['HOU', 905], ['ATL', 1255]], 3321: [['HOU', 1335], ['ATL', 1615]], 2133: [['BNA', 900], ['IND', 1115]]}, 'Southwestern': {2121: [['MDW', 430], ['SAT', 825]], 4335: [['PHX', 450], ['MDW', 745]], 1102: [['MDW', 1100], ['PHX', 1450]], 1111: [['SAT', 430], ['MDW', 825]]}, **'JET BLUE': {1010: [['IND', 605], ['MDW', 705]]},** 'Delta': {1776: [['PHL', 1350], ['RAP', 1610]], 1009: [['MDW', 1850], ['IND', 1955]], 1445: [['ATL', 1135], ['LAX', 1810]], 1096: [['PHX', 900], ['MDW', 1255]], 9001: [['MDW', 2145], ['IND', 2255]], 1226: [['PHX', 950], ['MDW', 1345]], 1885: [['ATL', 1305], ['LAX', 2000]], 1102: [['IND', 1850], ['MDW', 1955]]}}

4) Write a function **def reportFlightsByAirlineByFlightNumArrivePriorTo(flightsD,ArrivalTime):** which shows all flights that arrive before a specified time by airline (sorted) by flight number (sorted).

>>> reportFlightsByAirlineByFlightNumArrivePriorTo(flightsD,1200)

American NUM ARR TIME

2100 IND 1115

2133 IND 1115

Delta NUM ARR TIME

Southwestern NUM ARR TIME

1111 MDW 825

2121 SAT 825

4335 MDW 745

>>> reportFlightsByAirlineByFlightNumArrivePriorTo(flightsD,1600)

American NUM ARR TIME

1102 HOU 1450

2100 IND 1115

2133 IND 1115

4311 ATL 1255

5577 HOU 1350

Delta NUM ARR TIME

1096 MDW 1255

1226 MDW 1345

Southwestern NUM ARR TIME

1102 PHX 1450

1111 MDW 825

2121 SAT 825

4335 MDW 745

5) Write a function **def reportFlightsBySortedDepartureCityBySortedDepartureTime(flightsD):** that prints the following report:

>>> reportFlightsBySortedDepartureCityBySortedDepartureTime(flightsD)

ATL 1100

ATL 1135

ATL 1305

BNA 900

BNA 900

BNA 1100

HOU 905

HOU 1335

IND 1850

IND 1850

MDW 430

MDW 1100

MDW 1850

MDW 2145

PHL 1350

PHX 450

PHX 900

PHX 950

SAT 430

6) Write a function **def reportFlightsSameAirlineRoundTrip(flightsD,fromCity,toCity):** that reports any airline and flight for which a from/to exists and there also exists a corresponding return trip (from and to are reversed and with the same airline). BE SURE THE RETURN FLIGHT FROM THE TO CITY HAPPENS ONLY AFTER THE ARRIVAL!

>>> reportFlightsSameAirlineRoundTrip(flightsD,"IND","MDW")

American

Southwestern

Delta

1102 IND 1850 MDW 1955

9001 MDW 2145 IND 2255

>>> reportFlightsSameAirlineRoundTrip(flightsD,"PHX","MDW")

American

Southwestern

4335 PHX 450 MDW 745

1102 MDW 1100 PHX 1450

Delta

>>> reportFlightsSameAirlineRoundTrip(flightsD,"MDW","PHX")

American

Southwestern

Delta

7) Write a function **def reportFlightNumDupes(flightsD):** that will report on any airlines which have a flight number identical to another (not necessarily wrong, just could confuse a flier if they do not also look for the right airline company!) Be sure to only include the Airline Company names once in the list, e.g.

>>> reportFlightNumDupes(flightsD)

{1102: ['American', 'Southwestern', 'Delta']} ***#all three have flight 1102***

8) Write a function **def fromToVia(flightsD,fromCity,toCity):** that informs the user of all direct flights, *but if no direct flight exists*, finds a multiple flight route (again, arrival flights must arrive before departure from same location. Also note that the flights do not need to be through the same airline company).

>>> fromToVia(flightsD,"IND","MDW")

FROM: IND TO: MDW

Delta 1102 is a direct from IND to MDW

>>> fromToVia(flightsD,"BNA","MDW")

FROM: BNA TO: MDW

American 2133 BNA 900 IND 1115

Delta 1102 IND 1850 MDW 1955