

CS2401: Assignment #4

Due Date: Sunday, September 27, 2009.

Objective: The goal of this assignment is to practice recursion using 2-dimensional arrays.

Background: The table *flights* below is a 2-dimensional array with information about direct flights operated by the UTEP Airline. Each direct flight in the array is composed of two strings: the first string is the code of the airport of origin and the second string is the code of the destination airport. For instance, the first flight in the table is {"ELP","IAH"} meaning that the UTEP Airline operates flights from El Paso (airport code ELP) to Houston (airport code IAH). Note that the fact that UTEP Airline operates direct flights from El Paso to Houston does not imply that it operates direct flights from Houston to El Paso (in fact, the table does not include direct flights from IAH to ELP, i.e., {"IAH","ELP"}).

```
String[][] flights = { {"ELP", "IAH"}, {"ELP", "DFW"},  
                        {"ORD", "ELP"}, {"ORD", "DFW"}, {"ORD", "SFO"},  
                        {"DFW", "SFO"}, {"DFW", "DCA"}, {"SFO", "MIA"},  
                        {"IAH", "JFK"}, {"IAH", "DCA"}, {"DCA", "MIA"},  
                        {"MIA", "PHX"}, {"PHX", "DCA"}, {"JFK", "PHX"},  
                        {"JFK", "ORD"}, {"SFO", "DFW"} };
```

Flights with connections require one or more stops. For example, a flight with connection from El Paso (ELP) to New York (JFK) is represented by two or more ordered direct flights such as {"ELP","IAH"}, {"IAH","JFK"}. This flight can be represented as a 1-dimensional array of strings

```
String[] oneflight = {"ELP", "IAH", "JFK"};
```

Further, this 1-dimensional array can be rendered as "ELP->IAH->JFK".

In the context of this assignment, flights, whether direct or with connections, cannot have cycles. For example, neither {"DFW","DFW"} nor {"DFW","SFO","DFW"} are considered flights because they have a cycle: both arrays start and end at "DFW".

Assignment: Using the definitions above, write a java program that uses the flights table above to compute and print all possible flights (direct flights and flights with connections) starting at El Paso (i.e., having "ELP" as the airport of origin).

Hint 1: The program should print more than 30 flights originating at ELP. To test your program, you may use another airport of origin such as PHX that will generate fewer flights. For instance, you may expect an output with the following information for PHX:

PHX->DCA
PHX->DCA->MIA

Hint 2: You may need to use 1-dimensional arrays to store candidate flights. A candidate flight (an array computed by your program and derived from table *flights*) may be considered a flight if it does not include cycles (and you may be required to write a method that detects cycles in an array of strings).