**PJ 9 Report Your Name: Francisco Valadez**

**A. The following is my Java program:**

**// Please copy your Java program into here from your Eclipse window. The code must be colored.**

**// You must not copy Java program from your .java file since the code over there is not colored at all.**

**// You must not show screen prints here.**

import java.util.\*;  
  
import javax.xml.crypto.dsig.keyinfo.RetrievalMethod;  
  
public class Routes   
{  
 private static char flight[ ][ ] =   
 { // 10 rows by 2 columns char array  
 {'Y', 'Z'}, {'W', 'Y'}, {'W', 'S'}, {'S', 'T'}, {'T', 'W'},   
 {'P', 'W'}, {'P', 'R'}, {'Y', 'R'}, {'R', 'X'}, {'Q', 'X'}   
 };  
  
 public static int visited[] = {0,0,0,0,0,0,0,0,0,0};  
   
 public static void unvisitAll()  
 {  
 Arrays.fill(visited, 0);  
 }  
  
 public static void markVisited(char originCity)  
 {  
 for(int i = 0; i < flight.length; i++)  
 {  
 if(flight[i][0] == originCity)  
 {  
 visited[i] = 1;  
 break;  
 }  
 }  
 }  
  
 public static char getNextCity(char topcity)  
 {  
 for(int i = 0; i < flight.length; i++)  
 {  
 if((flight[i][0] == topcity) && (visited[i] == 1))  
 {  
 return flight[i][1];  
 }  
 }  
 return ' ';  
 }  
 public static boolean isPath(char originCity, char destinationCity)   
 {  
 Stack<Character> stack = new Stack<Character>();  
 char topCity, nextCity;  
 unvisitAll(); // clear marks on all cities  
 stack.push(originCity); // push origin city onto stack, mark it visited  
 markVisited(originCity);  
 topCity = (char)(stack.peek());  
  
 while (!stack.isEmpty() && (topCity != destinationCity))   
 {  
 // loop invariant: stack contains a directed path   
 // from the origin city at the bottom of the stack  
 // to the city at the top of the stack  
 // find an unvisited city adjacent to the city on   
 // the top of the stack  
 nextCity = getNextCity(topCity);  
  
 if (nextCity == ' ')  
 stack.pop(); // no city found; backtrack  
 else // visit city  
 {   
 stack.push(nextCity);  
 markVisited(nextCity);  
 } // end if  
 topCity = (char)stack.peek();  
 } // end while  
  
 if (stack.isEmpty())   
 return false; // no path exists  
 else   
 return true; // path exists  
  
} // end isPath  
  
 public static void main(String[] args)  
 {  
 int counter = 1;  
 char p1, p2;  
 Scanner input = new Scanner(System.in);  
 System.out.println("Welcome to the Searching Routes Game of Francisco Valadez!");  
 System.out.println(counter + "===========================================");  
 System.out.print("Enter the departure and arriving airports: ");  
 p1 = input.next().charAt(0);  
 p2 = input.next().charAt(0);  
 while((p1 != '@') && (p2 != '@'))  
 {  
 counter++;  
 //function here  
 if(isPath(p1, p2))  
 System.out.println("P1 and P2 are " + p1 + " and " + p2 + ". The answer is ......");  
 else  
 System.out.println("P1 and P2 are " + p1 + " and " + p2 + ". The answer is NO route available.");  
   
 System.out.println(counter + "===========================================");  
 System.out.print("Enter the departure and arriving airports: ");  
 p1 = input.next().charAt(0);  
 p2 = input.next().charAt(0);  
  
 }  
 System.out.println(counter + 1 + "===========================================");  
 System.out.println("Thank you for playing the Searching Routes Game of Franciso Valadez!");  
 System.out.println(counter + 2 + "===========================================");  
 }  
}

**B. The following is the complete output of my 5 test cases: [You must test those 5 test cases.]**

**// Please copy your Eclipse console output into here.**

**Program is stuck on infinite loop:**

