**Algorithm**

Based on the requirements, we have created three input text files that specify all the flight information for the airline as follows:

* Names of the cities that HPair serves
* Pairs of city names, each pair representing the origin and destination of one of HPair's flights
* Pairs of city names, each pair representing a request to fly from some origin to some Destination.

Let's create a graph with adjacent vertices and direct path. Figure 1.1 represents the graphical representation of a bi-directional tree. Consider A to J are the city names, we need to travel from A to J. ABCD gives the wrong guess1, pointer backtracks to A and starts the second guess – ABEFG which is again a false guess. Finally, path ABEFHIJ is going to be the correct path from origin to the destination. Figure 1.2 represents the matrix of the graph with edges and vertices where 0 is the false case(guess) and 1 is the true case(guess).

Figure 1.2

I

G

H

F

E

BB

A

Figure 1.1

C

D

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

J

**Algorithm**

1. Build graph with adjacent vertices and edges.

2. System takes origin and destination city names.

3. Map header file verifies if the city names are given in the input text documents.

4. If the city name doesn't match with the input file details then the output will be 'HPair doesn't fly to the city.

5. Else, pointer travels from origin to different nodes searching for the destination, taking different guesses.

6. If the destination is found, displays 'HPair serves from <origin> to <destination>

7. Else prompt error message.

**Pseudocode:**

Main()

Read given string

Read flight Details file

Verify UserReq file

bool isValidFlightRoute( initialize origin and destination as string, define a pointer to search flights TravelingPairs\* flights, int sizeOfFlights, map<string, bool> reached) {

if (origin is equal to destination) {

if (check debug) {

cout << Destination city is found, display the destination << endl

// Flight reached the destination city

return true

} else {

if (correct guess, both origin and destination are found in the flight list){

true case

} else {

if (check debug) {

cout << Backtrack to origin as the first guess is a failure << endl

}

return false;

}

for (initialize i and check all the flights from the flights file) {

if (origin is found) {

if (check debug) {

cout << display HPair travels from origin to destination << endl

}

if (check for the valid route from origin to destination){

return true;

}

}

}

if (check debug) {

cout << wrong guess, pointer reached the dead-end so backtracks to origin << endl;

}

//No flights

return false

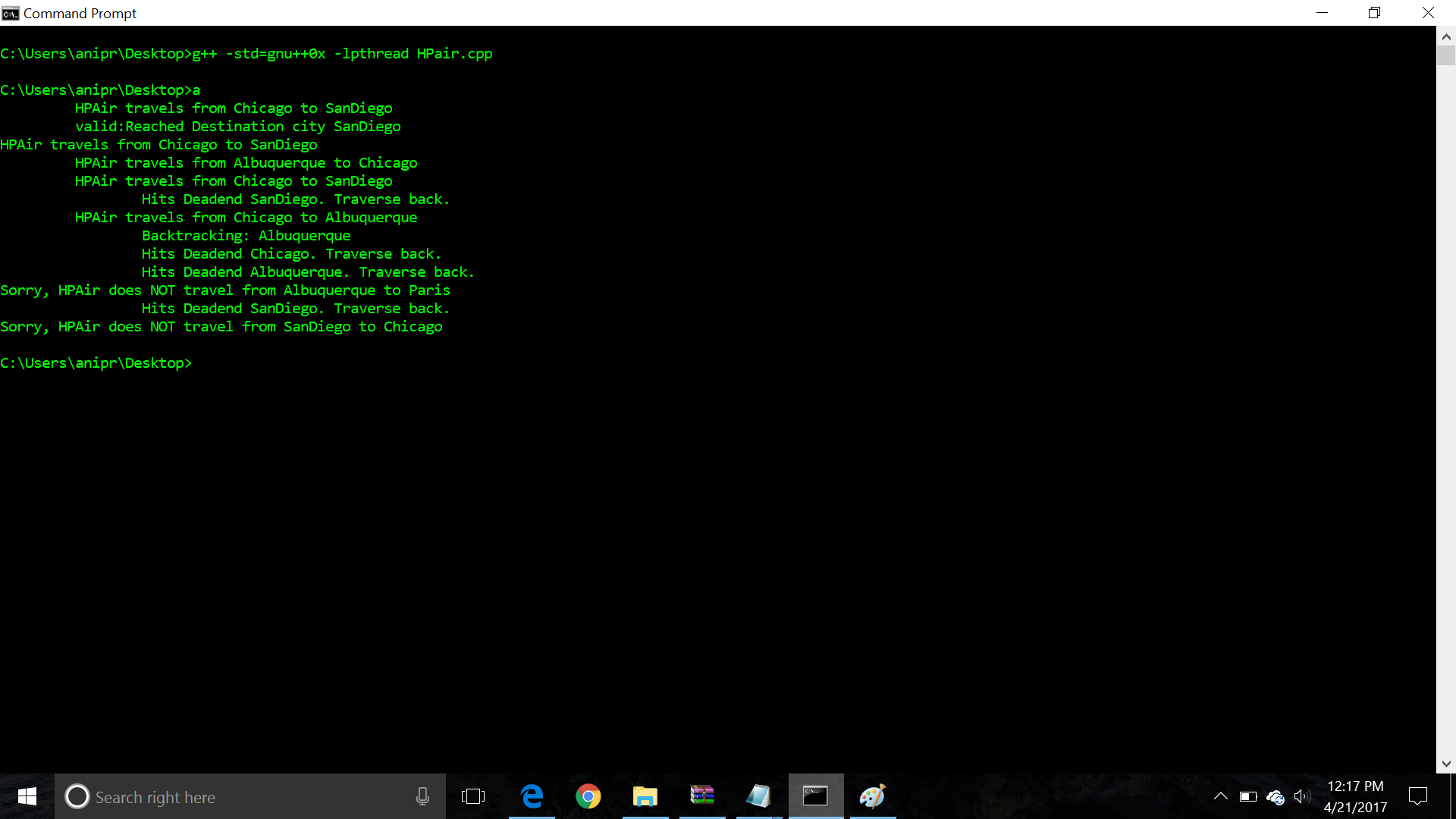
**Execution:** by using command prompt

g++ -std=gnu++0x -lpthread filename.cpp

**Output:**

1. True case

#define DEBUG true



2.False case

#define DEBUG false

