Flight Itinerary

Jack Webster
H00154950

Section 1 Testing

1.1 Program 1, Part A: Representing Direct Flights

Testing for part A was done using .toString() method on the graph, printing out each direct flight. Adding the flights to the graph and printing it as a string gives the result:

([Edinburgh, Heathrow, Amsterdam, Boston, Chicago, Montreal, Toronto, New Delhi, Shanghai, Hong Kong],
[(Edinburgh : Heathrow)=(Edinburgh,Heathrow),
(Heathrow : Edinburgh)=(Heathrow,Edinburgh),
(Heathrow : Amsterdam)=(Heathrow,Amsterdam),
(Amsterdam : Heathrow)=(Amsterdam,Heathrow),
(Heathrow : Boston)=(Heathrow,Boston),
(Boston : Heathrow)=(Boston,Heathrow),
(Boston : Chicago)=(Boston,Chicago),
(Chicago : Boston)=(Chicago,Boston),
(Boston : Montreal)=(Boston,Montreal),
(Montreal : Boston)=(Montreal,Boston),
(Montreal : Toronto)=(Montreal,Toronto),
(Toronto : Montreal)=(Toronto,Montreal),
(Edinburgh : Chicago)=(Edinburgh,Chicago),
(New Delhi : Shanghai)=(New Delhi,Shanghai),
(Shanghai : Hong Kong)=(Shanghai,Hong Kong),
(Hong Kong : Shanghai)=(Hong Kong,Shanghai)])

Here we see that printing the graph shows us each flight with the starting airport and finishing airport, as well as an array of all airports referenced.

1.2 Program 1, Part B: Least Cost Connections

Testing for part B was done by calculating the cheapest path of a journey and returning the direct flights as a string.

Test Data:

Depart: Edinburgh Arrive: Amsterdam

> [(Edinburgh : Heathrow), (Heathrow : Amsterdam)]

Here we see that the program can calculate a path from the first airport to the second using the direct flights declared in part A.

Depart: Edinburgh Arrive: Chicago

> [(Edinburgh : Heathrow), (Heathrow : Boston), (Boston : Chicago)]

Here we see that although there is a direct flight from Edinburgh to Chicago the program returns a different route, this is because the direct flight is more expensive than the suggested route, this shows that the program can successfully calculate the shortest route

1.3 Program 2, Part C: Additional Flight Information

The additional flight information is given when the directed flight is declared, using the ExtendedEdge class, and is shown through the Itinerary.

1.4 Program 1, Part D: Itinerary

Testing for part D was done by selecting a departure airport and an arrival airport, printing out the itinerary with the flight information and printing the total cost for the journey.

Test Data:

Depart: Montreal Arrive: Amsterdam

```
> Itinerary for Montreal to Amsterdam
        Leg     Leave           At      On      Arrive          At
        1       Montreal        2200    MA554   Boston          1110
        2       Boston          1410    BA123   Heathrow        2240
        3       Heathrow        2340    HA343   Amsterdam       0240
        Cost of journey: £430.0
```

Here we can see that the additional flight information from part C displays correctly and the program correctly calculates the total cost of the journey

1.5 Program 2, Part E: Itinerary Duration

Testing for Part E was done by selecting a departure airport and an arrival airport, printing out the itinerary and the total time in the air.

Test Data:

Depart: Montreal Arrive: Amsterdam

```
> Itinerary for Montreal to Amsterdam
        Leg     Leave           At      On      Arrive          At
        1       Montreal        2200    MA554   Boston          1110
        2       Boston          1410    BA123   Heathrow        2240
        3       Heathrow        2340    HA343   Amsterdam       0240
        Cost of journey: £430.0
        Air time: 24 hours and 40 minutes
```

Here we can see that the total time in the air has been calculated correctly and shows the accumulated time spent flying.

1.6 Program 1, Part F: Alternative Extensions

From the list of alternative extensions I have chosen to implement number 1 and number 2, as well as an extension of my own.

The first extension calculates the total journey time for the route, printing out the time taken for all the flights and time in between, however the extension does not convert destination arrival times to local time.

Test Data:

Depart: Montreal Arrive: Amsterdam

```
> Itinerary for Montreal to Amsterdam
        Leg     Leave           At      On      Arrive          At
        1       Montreal        2200    MA554   Boston          1110
        2       Boston          1410    BA123   Heathrow        2240
        3       Heathrow        2340    HA343   Amsterdam       0240
        Cost of journey: £430.0
        Air time: 24 hours and 40 minutes
        Journey time: 28 hours and 40 minutes
```

Here we can see that the journey time has been calculated correctly and shows the accumulated time from departing on the first flight to arriving on the last one.

The second extension calculates journey times for flights that span more than one day

Test Data:

Depart: Edinburgh Arrive: Amsterdam

```
> Itinerary for Edinburgh to Amsterdam
        Leg     Leave           At      On      Arrive          At
        1       Edinburgh       1130    EA115   Heathrow        1230
        2       Heathrow        2340    HA343   Amsterdam       0240
        Cost of journey: £210.0
        Air time: 4 hours and 0 minutes
        Journey time: 15 hours and 10 minutes
```

Here we can see that the program correctly handles the flight times and still produces correct times for air travel and the overall journey.

The third extension provides error handling when entering departure airports and arrival airports.

Test Data:

Type departing airport:	
> Glasgow	Entering non-existing airport
“Airport does not exist, please try again”	Program handles error
> Edinburgh	Entering existing airport
Type destination airport:	Program accepts input
> Berlin	Entering non-existing airport
“Airport does not exist, please try again”	Program handles error
> Edinburgh	Entering existing airport, identical to departing airport
“Departing from this airport”	Program recognises input and handles it
> Amsterdam	Entering existing airport
Itinerary for Edinburgh to …	Program accepts input

Here we can see that the program can identify which airports are valid inputs as well as recognising when the same airport is chosen for both departure and arrival

Section 2 Evaluation

Although the core functions of the program are working and produce correct results there are areas of the program that fail to fully or correctly carry out their intended function.

The first extension regarding the total journey time does not always produce correct results for the minutes and will often give values greater than 60. Although the values themselves are correct and add up to give the correct total time the minutes should be constrained to values less than 60 and the hours presented correctly. Given more time I would analyse the functions themselves to try and find the reasoning behind these issues and fix them.

The third extension does not completely cover all error handling scenarios and fails to handle being given two airports with no connections to one another, giving an error and stopping the program as a route could not be established. Given more time I would correct the issue and produce a suitable handler allowing the program to continue running and the mistake to be revoked without causing further issues.