Project 3 Report

I decided to develop incrementally until I got the first function working, by dividng it into 4 helper functions, getAirline, flightNumber, plusOrMinus, and getMinutes.

To consider alternate mutations of each airline code are you switch cases so that it would consider all of these. Then, for the flight number function, I used a temporary string in order to convert character to digit. Once I got this to work I used a similar code for the get minute function. The plus or minus was the simplest function because it only had to consider on two different types of characters. If the position that was being considered was not a plus or a minus, the function would quit and therefore be invalid. At this point I started to make the isWellFormedAirport string function. In order to do this I had to consider that if every single aforementioned helper function what is valid then it would be well formed. I did this by using nested if statements. After several assert statements I found that my function was working correctly.

Then, I created the next function, the ontime of arrival percentage. I first had to check if the airline code was in the commands string. If so, I coded to count each time that it appeared, to use later as the denominator for the ratio. Then, I used a while loop and if statements to count if the desired airline code was followed was a ‘-‘, it would count as a flight on time, and incremented. If a plus was found, and it is followed by a 0, or followed by a digit less than 15. Lastly, I wrote an equation that would calculate the on time percentage, and return the result as a floating point.

Pseudocode:

//getAirlineCode

//switch case

case A

increment position, if followed by A or L, consider valid.

Case S

…….

…

…

default

break // nullify any other inputs

// getMinutes and getAirlineCode

string temp

+=commands string

result, multiply by 10.

//plus or minus

if statement, if + or - , valid

else, not valid.

isWellFormedAirportString

if getMinutes and airlineCode and plusOrMinus and getAirline valid,

it is well formed airport string.

//ontimeArrivalPercentage

while loop

if commands position = to airline position

valid, increment for denominator.

if -, flight on time, increment.

If + and followed by zero or digit less than 15, flight on time increment.

Return equation result.

Significant Assert Statements:

*To test if well formed, and for it to return invalid if not:*

isWellFormedAirportString(“UA12+1”) … and all permutation of UA

isWellFormedAirportString(“AA123+12”) .. and all permutation of AA

isWellFormedAirportString(“DL123+12”).. and all permutation of DL

isWellFormedAirportString(“SW123+12”).. and all permutation of SW

isWellFormedAirportString(“AA123+12”)

! isWellFormedAirportString(“UA12-1”)

!isWellFormedAirportString(“Uvghajhdvj2+1”)

! isWellFormedAirportString(“aAa3%1A”)

isWellFormedAirportString(“UA123+12AA34+98DL76+2”)

!isWellFormedAirportString(“Sww21467587473”)

ontimeArrivalPercentage(“UA12+1”, “UA”) //test most basic strings:

ontimeArrivalPercentage (“AA123+12”, “AA”) .. and all permutation of AA

ontimeArrivalPercentage (“DL123+12”, “DL”).. and all permutation of DL

ontimeArrivalPercentage (“SW123+12”, “SW”).. and all permutation of SW

ontimeArrivalPercentage (“AA123+12”, “AA”)

ontimeArrivalPercentage(“UA123+12AA34+98DL76+2”, “UA”)

ontimeArrivalPercentage (“ua999-0aa1+15AA2+2”, “AA”)

ontimeArrivalPercentage (“"ua999-0aa1+15AA2+2", “DL”) //test if it doesn’t work for airline codes not found

ontimeArrivalPercentage ("ua999-0aa1+15AA2+2", “UA”) //able to search strings

ontimeArrivalPercentage (“AdefeggAsas123+12”, “AA”)//test if it doesn’t work for invalid strings

ontimeArrivalPercentage(“UA123+12aA34+98dL76+2UAUa”, “UA”) //able to search string and consider lowercase valid