**Project 3 Report**

Satvik Anand

404-823-011

Discussion 1E

1. **Obstacles overcome**

* I was unable to figure out how to make the program read the airline number and minutes correctly, but I solved that problem by using arrays.
* I found it difficult to make the program loop over multiple airline strings. I overcame this problem by using pass by reference to keep track of where the loop was in the string.
* After completing my code, it refused to initially run on g++. This is because my code had out of bounds errors. I eventually detected these errors and successfully remove them.

1. **Project description**

The following are the various functions in the project:

bool isWellFormedAirportString ( string commands ) ;

double ontimeArrivalPercentage ( string commands , string airlinecode ) ;

// retrieves the airline code

string airlineCode ( string commands , size\_t position ) ;

// checks the validity of the airline code

bool validAirlineCode ( string airlineCode , size\_t& position ) ;

// retrieves the airline number

string airlineNumber ( string commands , size\_t position ) ;

// checks the validity of the airline number

bool validAirlineNumber ( string commands , size\_t& position) ;

// checks whether there's a plus or a minus after the airline number

bool validPlusOrMinus ( string commands , size\_t& position ) ;

// retrieves the minutes

string minutes ( string commands , size\_t position ) ;

// checks the validity of the minutes

bool validMinutes ( string commands , size\_t& position ) ;

// counts how many individual flight codes are there

int counter ( string commands ) ;

// checks whether the airline code in the second function is valid

bool wellFormedAirlineCode ( string commands ) ;

// retrieves minutes of the early flights

string earlyMinutes ( string commands , size\_t position ) ;

// checks whether the minutes are early or late

bool early ( string commands , size\_t position ) ;

// counts the total number of flights

double flightCounter ( string commands , string airlinecode ) ;

// counts the number of early flights

double ontimeFlightCounter ( string commands , string airlinecode ) ;

// lowers the case of the values entered if it's a string of 2 letters

string decapitalizer ( string commands ) ;

The following are the functions and what they do

int main() {

}

bool isWellFormedAirportString ( string commands ) {

A loop that adds 1 to trueCount if each of validAirlineCode, validAirlineNumber, validPlusOrMinus and validMinutes are true.

If trueCount is divisible by 4 then it shows that there haven’t been any errors. If trueCount is divisible by 4, the function returns true, else the function returns false.

}

string airlineCode ( string input , size\_t position ) {

if position + 2 is less than commands.size(), and if the first two letters of the string at position are alphabets,

the function will return the string

}

bool validAirlineCode ( string input , size\_t& position ) {

  if the string returned by airlineCode is a valid airline code, function returns true, else the function returns false.

}

string airlineNumber ( string commands , size\_t position ) {

if position + 3 is less than commands.size(),

A loop which starts at position and goes upto position + 2. The loop stores the numbers collected in a string called airlineNumber.

}

bool validAirlineNumber ( string commands , size\_t& position ) {

Retrieves the string from airlineNumber and converts each digit into an integer.

Creates an integer which is a combination of all the letters of the integers collected.

If the integer is between 0 and 999, the function returns true, else the function returns false.

}

bool validPlusOrMinus ( string commands , size\_t& position ) {

If the character at position is a + or -, it will return true else it will return false.

}

bool validMinutes ( string commands , size\_t& position ) {

Stores the function produced by the minutes function in a string.

Converts every character of the string into an integer.

Constructs an integer which is a combination of all the above integers.

If the number is greater than 0 and lesser than 999, it returns true, else it returns false.

}

bool early ( string commands , size\_t position ) {

Stores the function produced by the minutes function in a string.

Converts every character of the string into an integer.

Constructs an integer which is a combination of all the above integers.

If the number is greater than 0 and lesser than 999, it returns true, else it returns false.

}

string minutes ( string commands , size\_t position ) {

If the character is a digit at position, position + 1 and/or position + 2, it stores the values of these characters in a string.

}

int counter ( string commands ) {

If there is a + or a – in the string, it adds 1 to integer count.

Function returns count.

}

bool wellFormedAirlineCode ( string commands ) {

If the airline code entered in the second function is valid, function returns true, else the function returns false.

}

string decapitalizer ( string commands ) {

Reduces to lower case, if any of the two letters in the second function’s airlinecode are upper case.

}

double flightCounter ( string commands , string airlinecode ) {

A loop, which adds to integer count if the decapitalized version of airline code is equal to the airline code in the string commands.

}

double ontimeFlightCounter ( string commands , string airlinecode ) {

A loop which adds to count every time a string is encountered, which satisfies that the airlinecode is equal to the airline code within commands.

Returns the number of on time flights.

}

double ontimeArrivalPercentage ( string commands , string airlinecode ) {

verifies that the string is well formed and the airline code entered is valid.

Calculates percentage of on time arrivals.

}

1. **Test data:**

Ensuring that everything valid works with the compiler:

* To test the airline codes, you need to enter all the valid airline codes in both capitalized and lower case forms, and a mix of both.

Eg: aa, AA, aA and Aa for all the airlines. So you can enter this in a valid string such as Aa12+43 or aa555-433.

* To test the airline numbers, you need to test one digit, two digit and three digit - all of them should work in a valid string.

Eg: Aa1+433, aA12+433, aA123+433.

* To test the plus or minus, you need to enter a valid airport string with a + or a – and it should work.
* To test the minutes, you need enter combinations of one digit, two digit and three digit, all of them should work in a valid string.

Eg: Aa1+1, aA12+12, aA123+123.

After this, you need to make sure that the above validities work over multiple codes, so you should enter multiple valid codes one after the other.

Eg: aa123+43sW33-544al109-443.

Ensuring that the compiler rejects invalid entries:

* You must enter invalid airline codes.
* You must enter airline numbers which are too long

Eg: aa1234+433

* You must enter no airline number at all.
* You must enter minutes which are too long

Eg: aa123+3333

* You must enter no minutes at all.
* You need to enter a blank string.

You need to make sure that these errors also occur over multiple codes.

Eg: aa123+4322aR123+433, aa123+433al1223+34, sw+43.