1:  
No1: The number can have different digit numbers, so have to write a switch function to react to situations with different digits to make sure that the following function works well.

No2: For the arrival percentage, it is hard to compare several digits in the string to 15, so I also use a switch function with different cases.

2:

addition function:

string toLower:

for(o to size of the string)

tolower every character of the string

return string

bool checkAirline(string a, int positionA):

create a string temp = a[positionA]+a[positionA+1]

if(temp= one of the airline code)

return true

else

return false

bool checkNumber(string b, int & positionB):

create temp=0

for(from position to size of the string)

if(isdigit()

temp++

else

break //find how many consecutive numbers are here

switch(temp) //change the location to start for next function

case 1:

positionB++;

case 2:

positionB +=2;

case3:

positionB+=3;

if(0<temp<4)

return true

else

return false

bool checkTime(string c,int positionC)

if( c[positionC]==’+’ or ‘-’)

return true

else

return false

bool isWellFormedAirportString(string commands)

create int position=0

tolower all the characters of the string

while(position < size)

first use checkAirline(command, position) check the first two characters

position +=2

then use checkNumber to check if the number is between 0 and 999 and also change the

value of position

then use checkTime to check if there is a ‘+’ or ‘-’

use checkNumber again

double ontimeArrivalPercentage(string commands,string airlineCode)

create total and late=0

if(the string is not a well-formed airport string)

return -1;

if(airlineCode is not a airline code0

return -1;

for all character

if there is two consecutive characters= airline code)

then if it connect with a “+”

check consecutive numbers again

if 1 digit-on time

2 digits- >=15- late++

3 digits -- late

total++

percentage= late\*1.0/total

return percentage

3:

For isWellFormedAirportString function

“a”: test how the function reacts to one character--- my function will terminate because I use assert to make sure that there is no undefined behavior.

“AA12+1”: test if the Capitalized Airline code can be identified

“dl3210+213”: test what if the number exceeds 999

“dl210c213”:test what if there is no ‘+’ or ‘-’

“dl213+2aa231-1”:test for code with multiple codes

“ad213”:test for incorrect airline code

For ontimeArrivalPercentage function

(“aa1+2aa23+134”,”aa”): test the normal situation of percentage

(“va213”,”aa”):test what if the code is not well-formed

(“va213-21al22+1”,”aa”):test what if there is no code for the airline code