ECOR 1606 DEF Winter 2013 - Assignment #3

Note: Program comments (including name / student number) and proper formatting (i.e. indentation) are required for full marks on all programs submitted in this course!

Question 1

Enhance your solution (or the sample solution) to assignment #2 part 2 ("*a22.cmm*") as follows:

When 0 0 is entered you program should output the following statistics:

i) The number of tries to get a valid room size
ii) The number of carpet sizes entered
iii) The number of invalid carpet sizes entered
iv) The number of valid carpet sizes entered
v) The number of carpets that fit exactly
vi) The number of carpets that can be trimmed
vii) The number of carpets that cannot be used
viii) The percent of invalid carpets
ix) The percent of valid carpets
x) The percent of valid carpets that fit exactly
xi) The percent of valid carpets that can be trimmed
xii) The percent of valid carpets that can be used (i.e. fit exactly or trimmed)
xiii) The percent of valid carpets that cannot be used
xiv) The average area of all of the valid carpets
xv) The area of the largest valid carpet
xvi) The area of the smallest valid carpet

As always, the statistics should only be calculated if it is reasonable to do so, e.g. make sure you never divide by 0.

Again, a sample executable has been provided. Your program should behave just like this one.

You can write this program ("*a31*") in C-- or in C++. However, you will submit a C++ file, "*a31.cpp*". If you prefer to work in C--, call your program "*a31.cmm*" and then use the "Create C++ File" option from the C-- "File" menu to create "*a31.cpp*".

Submit "*a31.cpp*" as Assignment 3; A#3 – Question 1.

Question 2

OC Transpo is doing a survey of its service. They have had somebody record the times at which #95 busses pass the Rideau Centre, and need a program which will analyze this data. You are to write a program that reads in times (in hours and minutes on a 24 hour clock) until 99 99 is entered. When 99 99 is entered, your program should output:

i) the number of valid and invalid times entered
ii) the shortest gap between any two busses
iii) the longest gap between any two busses
iv) the average gap length

The following sample run should give the idea:

```
Enter time in hours and minutes (99 99 to stop): 10 20
Enter time in hours and minutes (99 99 to stop): 10 5
** Time ignored - not >= previous time. **
Enter time in hours and minutes (99 99 to stop): 10 31
Enter time in hours and minutes (99 99 to stop): 55 1
** Time ignored - hours and/or minutes invalid. **
Enter time in hours and minutes (99 99 to stop): 10 39
Enter time in hours and minutes (99 99 to stop): 99 99

3 valid times and 2 invalid times were entered.
The shortest gap was 8 minutes long.
The longest gap was 11 minutes long.
The average gap length was 9.5 minutes.
```

Times may go past midnight. If a time entered is before 1 0 (1 am) and the previous time was after 23 0 (11pm), it should be assumed that times have gone from one day to the next.

If an entered time is invalid, your program must output an error message and ignore the time. Hour values must be between 0 and 23 (note: 24 0 is not a valid time), minute values must be between 0 and 59 and, except for the special case described above, times cannot be less than the previously entered time.

Note that it makes no sense to output gap information if fewer than two valid bus times have been entered.

Hint 1: It is strongly suggested that you convert times entered to "total minutes" before doing any processing.

Hint 2: Begin by writing a program that does not allow times to go past midnight (a simpler problem). Then, once you have this program working, modify it so that times can go past midnight.

A sample executable has been provided. As usual, you want to make your program behave exactly like the one provided. If any of the above is unclear, try running the sample.

Note that you may assume that the user enters only integer inputs, and it's ok if your C++ program "fails" (infinite loop in the sample exe) if the user enters real numbers, characters, etc. If you choose to work in C--, however, you have only "double" variables, so you will either need to check that the user has entered integers using function isInt, or you need to change the doubles to ints after converting to C++.

You can write this program ("*a32*") in C-- or in C++. However, you will submit a C++ file, "*a32.cpp*". If you prefer to work in C--, call your program "*a32.cmm*" and then use the "Create C++ File" option from the C-- "File" menu to create "*a32.cpp*".

Submit "*a32.cpp*" as Assignment 3; A#3 – Question 2.