ECOR 1606 Winter 2014 - Assignment #3

Notes:

- **Program comments (including name / student number in the first line) and proper formatting (i.e. indentation) are required for full marks on all programs!**
- **If you use a sample solution as your starting point in any question, please indicate this in your comments.**
- **Programs that do not compile and run in the Dev-C++ lab environment (ME4390/ME4377/MC5010) get a maximum of 25%.**

Question 1

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

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

i) The number of tries to get a valid lot size
ii) The number of house sizes entered
iii) The number of invalid house sizes entered
iv) The number of valid house sizes entered
v) The number of houses that fit
vi) The number of houses that do not fit because of lot line clearance
vii) The number of houses that do not fit because they are too small
viii) The number of houses that do not fit because they are too large
ix) The percent of houses that fit
x) The percent of houses that do not fit
xi) The percent of houses that do not fit because of lot line clearance
xii) The percent of houses that do not fit because they are too small
xiii) The percent of houses that do not fit because they are too large
xiv) The average area of all of the houses that fit the lot
xv) The area of the largest house that fits the lot
xvi) The area of the smallest house that fits the lot

As always, the statistics should only be calculated if it is reasonable to do so, e.g. make sure you never divide by 0. If a house does not fit the lot clearance and also does not meet the either of the size requirements, it should only be counted in the lot line clearance count.

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

A survey to measure the activity level of residents of an assisted living facility was started in Jan 1, 2010. Each resident is asked to record the dates when they have gone out of the facility to shop, visit friends or family, or participate in a social engagement. The recorded dates are available for

each resident and they need a program which will analyze this data. You are to write a program that reads dates (as *year month day* where month is represented numerically: 1=January, 12=December) until 0 0 0 is entered.

When 0 0 0 is entered, your program should output:

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

The following sample run should give the idea:

```
Enter date as year month day (0 0 0 to stop): 2013 12 30
Enter date as year month day (0 0 0 to stop): 2013 12 31
** Date ignored - date invalid. **
Enter date as year month day (0 0 0 to stop): 2014 1 15
Enter date as year month day (0 0 0 to stop): 2014 1 12
** Date ignored - not >= previous date. **
Enter date as year month day (0 0 0 to stop): 2014 2 27
Enter date as year month day (0 0 0 to stop): 0 0 0
3 valid dates and 2 invalid dates were entered.
The shortest gap was 15 days.
The longest gap was 42 days.
The average gap length was 28.5 days.
```

Years need to be 2010 or later, months need to be between 1 and 12, and dates need to be between 1 and 30 (see Simplification below) except for the sentinel values (0 0 0).

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

Hint: It is strongly suggested that you convert dates entered to days since the project started before doing any processing.

Simplification: Your program **must** assume that all months have 30 days (and therefore all years have 360 days). Do **not** include the logic to manage months with 28, 29 or 31 days, or leap years.

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 *double*s to *int*s 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.