COMP 1602: Computer Programming II
Assignment #1
Date Due: **February 4th, 2018 at 11:55pm**

Instructions

- Create and test the program as detailed below. The name of the C++ file is your student id number.
- Ensure that the correct `.cpp` file is in the folder.
- Upload only the folder to myeLearning no later than **4th February, 2018 at 11:55pm.**
- Absolutely _no late_ or _emailed_ submissions will be accepted.

Program Description

It is important for persons with diabetes to check their blood sugar level on a daily basis. Such persons should aim for a blood sugar level between 70 mg/dl to 130 mg/dl before meals.

The file, `readings.txt`, contains the blood sugar level of a person with diabetes taken **each** day for several months (one reading per day). The amount of values in the file is unknown beforehand but a value of -1 indicates that there is no more data.

Write a program with the following functionality:

(a) Read all the data from the file and store it in an array, _glucose_. There are at most 500 readings in the file. Generate an error report to an output file, `errors.txt`, for any blood sugar value that is less than 20 or greater than 400 (do not store these values).

(b) Find and display the day when the blood sugar level was the highest; if there are ties, all of them must be displayed.

(c) Find and display all the occasions (if any) when there were three consecutive days where the blood sugar level went above a certain number _n_, where _n_ is input by the user at the keyboard.

(d) A blood glucose level of zero on any day indicates that the person did not measure his/her blood sugar level on that day. Find and display the amount of days and the percentage of days the person did not measure his/her blood glucose level.

(e) A _spike_ is defined as a sudden **increase** or **decrease** in a person's blood sugar level by at least 40 mg/dl in two consecutive days, provided that a measurement was taken on both days.

Find and display the amount of spikes (if any) in the person's blood sugar level. Also, find and display the biggest spike and indicate between which two days it occurred.

(f) A sequence of numbers is said to be **strictly increasing** if each number is greater the previous one, e.g., 105, 110, 115, 120, 125, 130. A sequence is said to be **strictly decreasing** if each number is less than the previous one, e.g., 130, 125, 120, 115, 110, 105.

Given a *start* location and an *end* location in the array which are input by the user at the keyboard, determine if the blood sugar levels were strictly increasing, strictly decreasing or neither within the *start* and *end* locations. You should ensure that the *start* and *end* locations are valid locations in the array.

(g) Sometimes, during weekends, the blood sugar level of a person with diabetes may exceed the recommended maximum value due to eating habits. Find and display the amount of weekends in which this may have happened and the amount of weekends analyzed.

Note that the seventh value from the file was the measurement for the first Saturday. Seven days after that was the measurement for the following Saturday, and so on.

(h) For the seven ranges 70-79, 80-89, 90-99, 100-109, 110-119, 120-129, ≥130, draw a histogram showing the number of times the blood sugar level fell within each range.

Output one asterisk ("*") for every 5 times the blood sugar level was in that range. Fractional values should be rounded up using the *ceil* function.

The histogram should be formatted as follows (the actual shape would depend on the data):
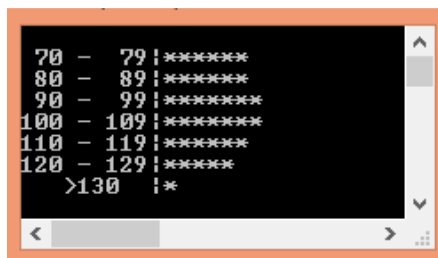
Programming Guidelines

1) The data file for the assignment is `readings.txt`. It is available for download at the course Web site in myElearning.

2) For parts (b) to (g) all output is written to an output file `results.txt`.

3) For part (h):
   a. Consider using another array where each element in the array represents a range (e.g., element 0 represents 70-79) and the value stored is the amount of times the blood sugar level was within the range represented.
   b. You should write a `printStars` function which accepts an integer parameter `n` and prints `n` stars on one line followed by the newline character.
   c. The output for (h) is written to the screen.