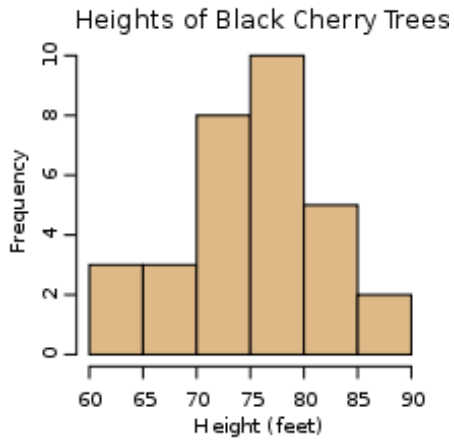
Morgan State University
Department of Electrical & Computer Engineering

EEGR 409: C Programming Applications

Project 1

Objective:
A botanist employs a recording device to help catalog measurements while out in the field. While the device can store individual measurements, for example, heights of Black Cherry trees, the botanist needs a quick way to generate histograms of the measurements on the device. A histogram is a graphical representation showing a visual impression of a distribution of data. A histogram consists of tabular frequencies, shown as adjacent rectangles, erected over discrete intervals (bins), with an area equal to the frequency of the observations in the interval. An example of a histogram for the height of 31 black berry trees is displayed in the figure below. You can find more information about histograms at www.mathsisfun.com/data/histograms.html or on Wikipedia.

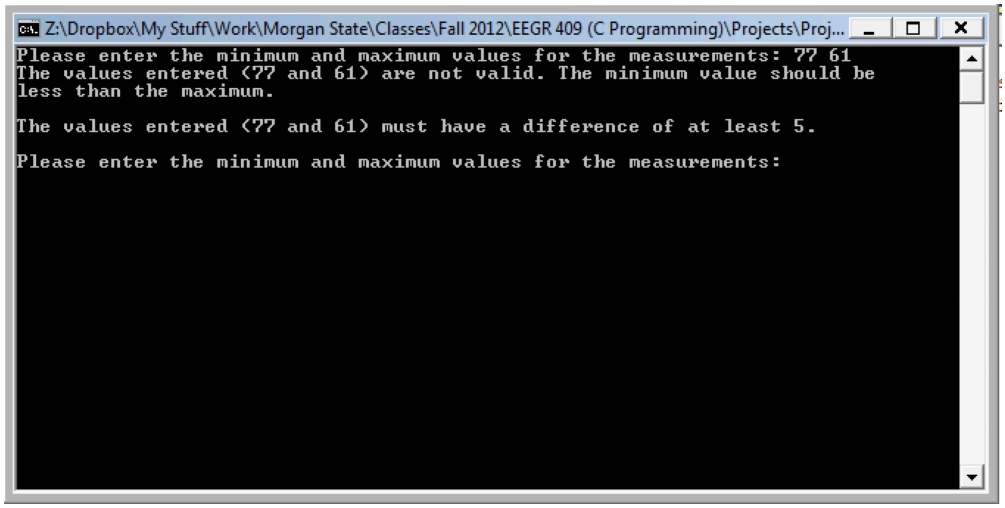


Write an application that will allow the botanist to enter a series of measurements and then display a **5-bin histogram** of those measurements. It is recommended that you complete this project in phases.

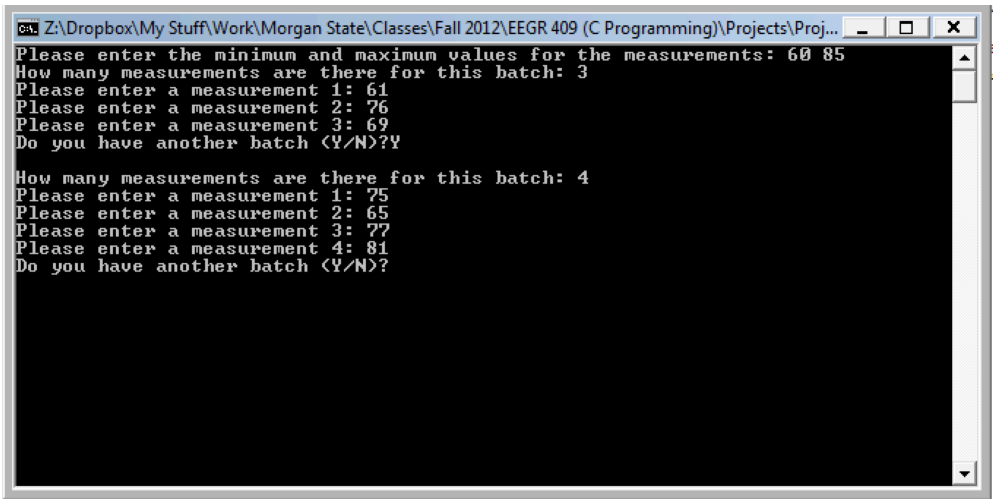
Phase 1: Get Measurements (Data)
Begin by asking the user to enter the range for the expected measurements. For the example of the heights of Black Cherry trees, the minimum and maximum of the range of measurements is 60 and 90 respectively. This information will be used to determine the range (size) of each bin for the histogram. Make sure you perform **input validation**:

1. Minimum of the range can't be greater than or equal to the maximum of the range
2. The difference between the maximum and minimum value of the range should be more than 5

If any of these two criteria is not met, the user is asked to try again.



Data can be entered in batches to save time. Ask the user for the number of measurements available for the current batch. Once again, perform **input validation**: the number of measurements per batch must be at least 1. Next ask the user to enter input measurements, and then repeat for the number of measurements in the batch. **Input validation** must be performed on each measurement: the measurement must fall in the range between the expected minimum and maximum. After a batch of measurements have been recorded, ask if another batch of measurements will be entered. Since we are not storing all the data values in arrays yet, for the time being you can simply discard each measurement after its been read. *Hint: remember to clear the input buffer after every input.*

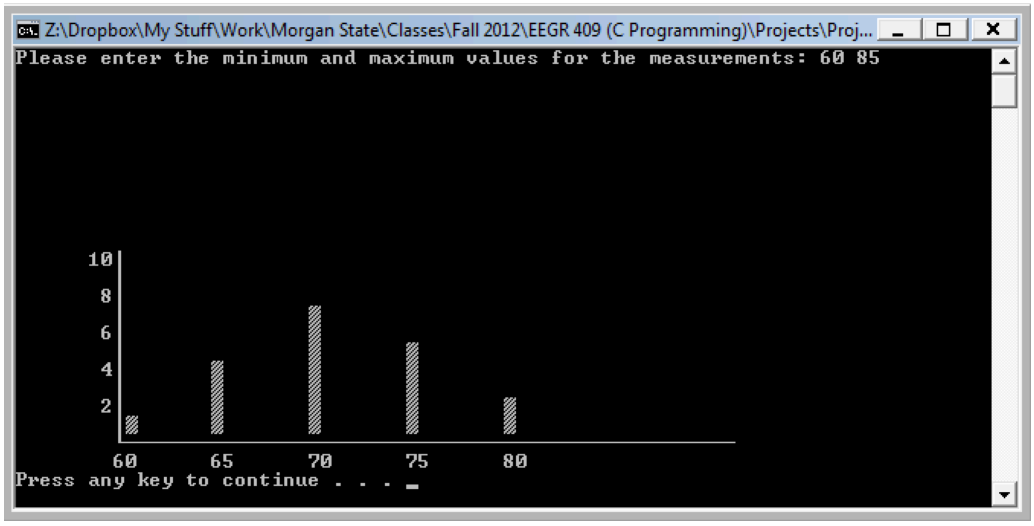


Phase 2: Implement Histogram Bins

After the user enters each measurement, you need to make a decision of which bin that measurement belongs to. Based on the expected minimum and maximum value, figure out how large each bin will have to be and then increment the bin counts as measurements are read.

Phase 3: Display Histogram Plot

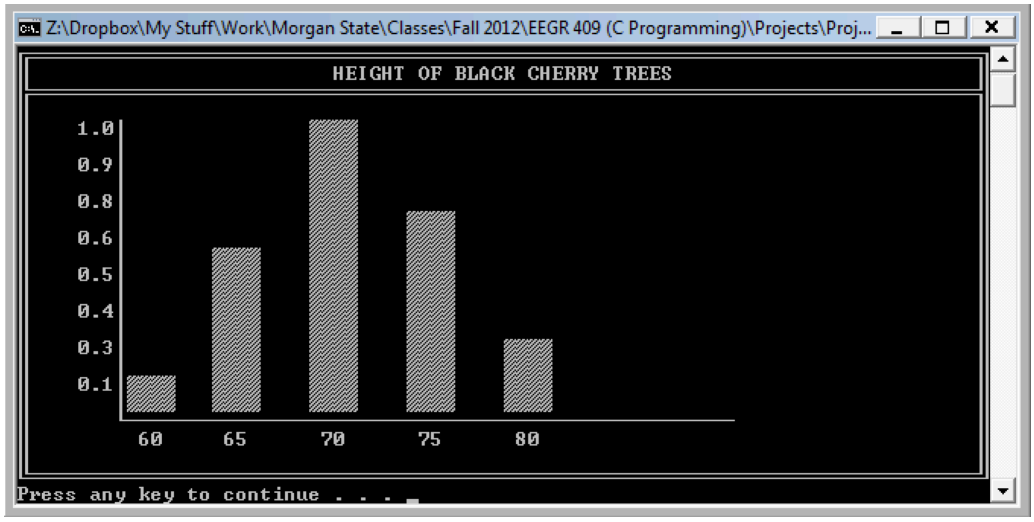
The y axis of the histogram plot should include only even numbers. The x axis should be defined by the expected minimum and maximum values for the range. Finally use ASCII characters to display the vertical and horizontal lines for the graph (and the boxes for the histogram bars). When complete, your sample data should be displayed as follows:



Once you are able to get the appropriate graph, uncomment out your previous code so the entire program works as displayed in the video on Black Board.

Extra Credit (+50 Pts):

1. Create a border around the graph and add a graph title (see figure below)
2. Graph bars should be 4 characters thick
3. Graph should take up most of the screen and the y axis should be normalized (between 0 and 1)



Functions:

Your program should create and make use of the following functions:

- *GetNumMeasurements()* – Returns the validated number of measurements for a batch
- *GetMeasurement* – Returns a single validated measurement from the user
- *UpdateBin* – Returns the updated value for a given bin based on a measurement (this function should be called 5 times, once for each bin)

The use of global variables are not allowed!

Grading:

1. Proper use of functions and input validation (20 pts)
2. Read data in batches – Phase 1 (30 pts)
3. Correctly place data into histogram bins – Phase 2 (20 pts)
4. Correctly produce a histogram graph – Phase 3 (30 pts)
5. Create border around the graph and add title (+10 pts)
6. Make sure graph bars are 4 characters in width (+10 pts)
7. Graph takes up the whole screen and is normalized (+30 pts)