

CSC 2430
Lab #2 – Array Quiz Data Bar Chart Analysis
Due: Wednesday, January 20, 2016

The purpose of this assignment is to improve your C++ programming proficiency in working with arrays, functions and report generation, and to give you practice working with Visual Studio. Be sure to implement a clean, well-designed, and commented C++ program solution.

Design, implement and test a C++ program using Visual Studio that inputs and stores in a single-dimension array a set of data values representing Quiz scores and the number of students that received each score. The input data consists of a collection of pairs of data values:

- the first integer value in each pair is a quiz score value (values between 0 – 50, e.g., 51 possible quiz score values)
- the second integer value in each pair is a frequency count of the number of students that received that quiz score value (e.g., frequency count values are ≥ 0)

(that is, array `ScoreFrequency[k]` is the number of students that received quiz score `k`).

The input data set is a variable-length sequence of these pairs of data values, with the end of the data set denoted by reading in a pair of sentinel data values where the quiz score value is less than 0 (and the associated frequency value is not significant, although it must be specified).

For example, a very small input data set might look like:

```
35 5
50 2
41 6
42 3
48 4
-1 0
```

which would result in the `ScoreFrequency[]` array of 51 values being all zeros except for the indicated array locations for subscripts 35, 41, 42, 48, and 50 having the associated array values indicated. If the data set contains more than one pair of (score, frequency) values for the same quiz score, then add the additional frequency value to the appropriate array entry (for example, if this data set also contained another pair of values " 42 2 ", then the `ScoreFrequency[42]` entry would be 5, \rightarrow the initial 3, plus the additional 2 from the second pair of values for quiz score 42).

Once the data is input and stored in the array, analyze the dataset and determine the lowest quiz score that any student received, the highest quiz score that any student received, and the largest frequency count for any of the quiz scores:

(for the small example, `lowestscore = 35`, `largestscore = 50`, and `largestfrequency = 6`).

Then produce a console output report with three sections:

1. First, print out the array values in a table format that identifies the quiz scores and associated frequency counts. Only output the sequence of array entries from the `lowestscore` up to the `largestscore` range. Also, output the 3 values determined in the analysis step.
2. Then, output a readable "bar chart" graph in a vertical arrangement down the page from the `lowestscore` to the `largestscore`, with each labelled line consisting of a horizontal bar string of asterisk characters ('*') representing the frequency count.

- Finally, output another readable "bar chart" graph in a horizontal arrangement where the quiz score values (from lowestscore to largestscore) appear across the page and the asterisk-bars are displayed vertically down the page on consecutive lines.

Example output (one long report, split up for printing in this lab assignment handout):

```
Welcome to the Quiz Score Frequency Analyzer, by Mike Tindall

Enter a list of pairs of values: "QuizScoreValue ScoreCount".
Example: 35 5 indicates 5 more students received a score of 35
Enter "-1 0" when finished:
35 5
50 2
41 6
42 3
48 4
42 2
-1 0

---Input Data---
Score: Frequency

    35:    5
    36:    0
    37:    0
    38:    0
    39:    0
    40:    0
    41:    6
    42:    5
    43:    0
    44:    0
    45:    0
    46:    0
    47:    0
    48:    4
    49:    0
    50:    2

The smallest score value is 35
The largest score value is 50
The largest frequency count is 6

Score: Frequency Vertical Bar Chart

    35: *****
    36:
    37:
    38:
    39:
    40:
    41: ******
    42: *****
    43:
    44:
    45:
    46:
    47:
    48: ****
    49:
    50: **

Frequency: Score Horizontal Bar Chart

    ^ 6:          *
    ^ 5:    *      * *
    ^ 4:    *      * *
    ^ 3:    *      * *
    ^ 2:    *      * *
    ^ 1:    *      * *
    -----:  --^--^--^--^--^--^--^--^--^--^--^--^--^--^--^--^
    Score: 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

Turn in printed assignment in class:

- Print out of source code (use Landscape orientation and a fixed-pitch font).
- Print out of sample execution run using the following test case data (careful – be accurate):
(note: cin handles free-format input streams and ignores line boundaries)

35 5 50 2 41 6 42 3 48 4 42 2 36 1 37 3 39 1 43 8 44 12 45 4 46 5 47 9 49 6 -1 0

- For the execution-run printout, you can either copy a screen shot of the execution window, or better yet, copy the text from within the execution command window using the built-in > Edit "Mark" mechanism that is available if you click the icon in the top-left corner of the command window. Scroll back to the top of the command window; select >Edit "Mark"; hold down the SHIFT key and use the arrow keys to move right and down to highlight the text you want to extract/copy; press "Enter" to do the copy onto the clipboard; then use normal "Paste" editing in NotePad or WORD. Be sure to format text in WORD using Courier New fixed-pitch font.