CSCE 240 (sections 2 and 3) – Programming Assignment Three

Due: 11:59pm on Tuesday, February 28th

Program Purpose

Read a text file and output:

- The alphabetic character(s) that appear most frequently in the file (note: for all counts, ignore the character's case. e.g. treat 'a' and 'A' as the same character)
- The alphabetic character(s) that appear least frequently in the file
- A list of the characters and their frequencies, sorted from most frequent to least frequent. Characters that appear with the same frequency should be sorted alphabetically
- A bar chart displaying the frequencies of each character with the bars displaying with asterisk. The horizontal axis should be the characters from a-z with a space between each character

Example Input / Output

Input file contents:

The quick brown fox jumps over the lazy dog.

Program's corresponding output:

Highest frequency character (appeared 4 times in the file): o
Lowest frequency characters (appeared 1 time in the file): a, b, c, d, f, g, i, j, k, l, m, n, p, q, s, v, w, x, y, and z

```
o: 4
e: 3
h: 2
r: 2
t: 2
u: 2
a: 1
b: 1
c: 1
d: 1
f: 1
g: 1
i: 1
j: 1
k: 1
l: 1
m: 1
n: 1
p: 1
q: 1
s: 1
v: 1
w: 1
x: 1
y: 1
z: 1

4                         *
3         *               *
2         *     *         *     *   * *
1 * * * * * * * * * * * * * * * * * * * * * * * * * *
  a b c d e f g h i j k l m n o p q r s t u v w x y z
```

Specifications

- All output should be directed to the standard output device using cout.
- Output formatting should match the example above and the sample output files given for the tests provided. Note the proper use of singular and plural (character/characters, time/times) and use of commas and "and" when listing characters.
- Your main function's signature should be: **main(int argc, char * argv[])**
- Your program will be run with the program name followed by the name of the input file your file should open and read (argv[1]). For example, if one types
  ./program3 test1-input.txt
  at the command prompt, program3 should open test1-input.txt from the current directory, read all of the text in the file, and produce output to the standard output device (using cout)
- Your main function must be implemented in *program3.cc*
- Place the function prototypes for all functions you write to call in your main in *program3functions.h*
- Place the function implementations for all functions you write to call in your main in *program3functions.cc*
- The only header files that may be included in your code are: iostream, fstream, iomanip, cmath, cctype, and program3functions.h
- You will submit a zip file (only a zip file will be accepted) containing *program3.cc*, *program3functions.h*, and *program3functions.cc*
- Programs must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).
- Be sure to review the program expectations section of the course syllabus.

Testing

Text files containing sample input and the corresponding expected output for *program3* are also attached to the assignment. A makefile has been included to run your program with the sample input and compare the results to the expected output. In order to use the makefile, ensure that your program3functions.h, program3functions.cc, and program3.cc files and all of the files attached to the assignment (*checkit.cc*, *correct-test1.txt*, *correct-test2.txt*, *makefile*, *test1-input.txt*, *test2-input.tx*t) are in the same directory. Your program will be graded using this same method with different input/output file pairs.

To run the included tests on your program, type:

```
make program3test1
make program3test2
```

Note: Differences in capitalization or spacing (including extra whitespace at the end of the output) and prompts for input will cause the provided tests to fail. End your last output statement with endl. The tests will display your output up to the first character that doesn't match the expected output. You can view your full output in the *student-test#.txt* file and compare it to the corresponding expected output in the *correct-test#.txt* file.

Grade Breakdown

Style: 1 point
Documentation: 1 point
Clean compile and link of *program3.cc*: 1 point
Correctly displays highest frequency characters for instructor tests: 1 point
Correctly displays lowest frequency characters for instructor tests: 1 point
Correctly displays character frequencies for instructor tests: 3 points
Correctly displays bar chart for instructor tests: 2 point

The penalty for late assignment submissions is 10% per day up to three days after the assignment due date. No assignment submissions will be accepted more that 3 days after the due date.