Assignment 5

Submit all the deliverables to CourSys: https://courses.cs.sfu.ca/
10% penalty per day late; assignments not accepted after 2 days late (20% penalty).
This assignment may be done individually or in pairs. Do not show non-group members your code, do not copy code found online or from previous course offerings, and do not post questions about the assignment online. Direct all questions to the instructor and TA.
See the marking guide for details on how each part will be marked.

1. Vector Statistics

Create a program named vectorStats.cpp which implements the following functions. You need not use function prototypes. You may use parts of lab solutions if applicable.

1.1 User Input

These three functions are the only functions in part 1 of the assignment which are allowed to use cout and/or interact with the user. All other functions may not use cout. Of course, you may use cout to help you debug, but your final assignment must not have those functions calling cout.

1. Read vector of floats:
   - This function prompts the user and reads in floating point numbers from the keyboard, storing them into a vector. A zero (0) input value is the sentinel which stops the input and is not put into the vector.
   - Return the vector of floating point numbers which was created in this function.

2. Display Numbers:
   - Passed a vector of floating point numbers and prints them to the screen.
   - Requirements for output formatting:
     1. Each line starts off with a tab indent.
     2. Each number printed with two decimal places and lining up on the screen in columns
     3. Numbers are comma separated; no trailing comma after last number.
     4. Display at most 8 numbers on a row.
   - Sample output for numbers: 1 2 3 4 5 6 7 8 9 10 11 12 13 100 100.1
     100.11 987.654321 1234.44 0.1

   1.00,   2.00,   3.00,   4.00,   5.00,   6.00,   7.00,   8.00,
   9.00,  10.00,  11.00,  12.00,  13.00, 100.00, 100.10, 100.11,
   987.65,1234.44,   0.10

3. Display Statistics:
   - Displays most of the output for the program. It will call the functions you'll write next.
   - Update this function to generate more and more of the target output as you write the later functions.
4. **Your main() function must be exactly:**

```c
int main()
{
    // Read in the values:
    vector<double> userNumbers = readVectorOfDoubles();

    // Display the output
    displayStatistics(userNumbers);

    return 0;
}
```

Create the functions described by the following sections:

### 1.2 Detecting Order

1. **Is Ascending**
   - Accept a vector of floating point numbers containing zero or more numbers.
   - Return `true` if the numbers in the vector are all in strictly ascending order; `false` otherwise.
   - Example: The vector {-1, 2, 5, 100} is ascending, but {1, 1, 5, 100}, {100, 5, 200} are not.
   - Note: The vector {5} is in ascending order, as is {} (empty vector).

2. **Is Descending**
   - Accept a vector of floating point numbers containing zero or more numbers.
   - Return `true` if the numbers in the vector are all in strictly descending order; `false` otherwise.
   - Example: {100, 3, -41, -50} is descending, but {1, 1, -5}, {100, 5, 200} are not.
   - Note: The vector {5} is in descending order, as is {}.

### 1.3 Calculating Average, Minimum and Maximum

1. **Average**
   - Pass in the vector of floating point numbers.
   - Return the average value of the vector. Return 0 if the vector size is zero.

2. **Minimum**
   - Pass in the vector of floating point numbers.
   - Return the minimum value in the vector. Return 0 if the vector size is zero.

3. **Maximum**
   - Pass in the vector of floating point numbers.
   - Return the maximum value in the vector. Return 0 if the vector size is zero.
1.4 Counting and Removing Duplicates

1. Count number of times a value is present
   - Pass in a vector of floating point numbers, as well as a single number named `value`.
   - Return how many times `value` occurs in the vector.

2. Get the most common value
   - Pass in a vector of floating point values.
   - Return the value which is found most often in the vector. In the event of a tie, you may return any of the values occurring equally most often.
   - Return 0 if the vector size is zero.

3. Remove Duplicates
   - Pass in a vector of floating point numbers by reference.
   - Remove from the vector any duplicate values. i.e., after calling this function, the vector argument in the calling code is changed, leaving in the vector only one occurrence of each number.
   - For example, given the vector `{4 1 2 1 2 1 3}` the calling code's vector ends up holding either `{4 1 2 3}` or `{4 2 1 3}` (it does not matter which of the duplicate values you remove, just that only one occurrence of each value is left).
   - Hint: From a vector `myVect` you can remove the element at index `i` with:
     ```
     myVect.erase(myVect.begin() + i);
     ```
   - Hint: When you erase (remove) an element in the middle of a vector, all elements after that removed element effectively move up in the vector. Hence if you removed element at index 5, the next element will now be at index 5. Be careful to handle this correctly and test with consecutive duplicates, like: `{1 1 1 1 1 1}` gives `{1}`.
1.5 Sample Output
Your program should generate the following output. Your format need not exactly mirror this, but it should be close.

You need not use any ANSI codes for this part.

<table>
<thead>
<tr>
<th>Enter value (0 to end):</th>
<th>1.2  2.8 1.2 5 1.2 1.2 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of the numbers:</td>
<td></td>
</tr>
<tr>
<td>Values:</td>
<td></td>
</tr>
<tr>
<td>1.20,  2.80,  1.20,  5.00,  1.20,  1.20</td>
<td></td>
</tr>
<tr>
<td>Statistics on those numbers:</td>
<td></td>
</tr>
<tr>
<td>Maximum:  5.00</td>
<td></td>
</tr>
<tr>
<td>Minimum:  1.20</td>
<td></td>
</tr>
<tr>
<td>Average:  2.10</td>
<td></td>
</tr>
<tr>
<td>Is ascending:  No</td>
<td></td>
</tr>
<tr>
<td>Is descending: No</td>
<td></td>
</tr>
<tr>
<td>Data with duplicates remove:</td>
<td></td>
</tr>
<tr>
<td>1.20,  2.80,  5.00</td>
<td></td>
</tr>
<tr>
<td>Repetition counts in user data:</td>
<td></td>
</tr>
<tr>
<td>Most common value is 1.20!</td>
<td></td>
</tr>
<tr>
<td>The value 1.20 occurs 4 time(s).</td>
<td></td>
</tr>
<tr>
<td>The value 2.80 occurs 1 time(s).</td>
<td></td>
</tr>
<tr>
<td>The value 5.00 occurs 1 time(s).</td>
<td></td>
</tr>
</tbody>
</table>

- For the ‘repetition counts in user data’, display the count for each unique value found in the vector. i.e., for each of the values in the “duplicates removed” vector, print out how many times that value was found in the user’s original input, not the duplicates removed vector.

- Hint: you’ll likely need to make a copy of the user’s original input before removing duplicates so that you have it here to analyze.
2. Letter Counting

Write a program named `letterCount.cpp` which allows the user to enter a line of text, and then counts and displays the number of times each letter appears in the text. For this part of the assignment you must use function prototypes and put all your function implementations below `main()`.

1. Read in a text
   - In `main()`, read in a full line of characters into a string.
   - Hint: Use `getline()` to read in text:
     ```cpp
     string x; // Use a better name!
     getline(cin, x);
     ```

2. Count the letters
   - Create a function which counts the occurrences of each letter.
   - It will need the user’s text to do this.
   - Return a vector of `int` of size 26 which counts the occurrences of each letter.
     For example, its first element (index 0) counts the number of a's and A's, ...
   - Hints:
     - You can convert a letter to upper case by calling `toupper()`. This function is in the header file `cctype`. Look it up online to determine its arguments.
     - A character is just a number. To find out which index an uppercase letter is, use:
       ```cpp
       int daIndex = myLetter - 'A';
       ```
     - Make sure you exclude non-letters from your count. Don’t count spaces, punctuation, numbers, or anything that’s not an a-z or A-Z.
     - Your count will be case insensitive: ‘G’ is counted the same ’g’.

3. Print the letter counts
   - Write a function which, give the occurrence counts you have already computed, prints a summary to the screen.
   - Use ANSI codes to colour each occurrence count displayed as follows:
     - Between 0-3 coloured in blue
     - Between 4-6 coloured in cyan
     - Between 7-10 coloured in bold cyan
     - 11 and over coloured in bold yellow

4. Your function implementations must be below `main()`. Put function prototypes above `main()` so that `main()` can call the functions as needed.
2.1 Sample Output
Some text, with a bunch of symbols and punctuation (not showing colours).

Enter your string.
: Hello world! This is the start of, well... um... Yeah! !@@@@@@&(!
You entered: Hello world! This is the start of, well... um... Yeah! !@@@@@@&(!

Character Count:
----------------


All characters.

Enter your string.
: The quick brown fox jumps over the lazy dog!
You entered: The quick brown fox jumps over the lazy dog!

Character Count:
----------------


No input (just hit enter).

Enter your string.
:
You entered:

Character Count:
----------------

(more on next page)
**Shakespear's Romeo and Juliet balcony scene**: Input text is all one long line, just word-wraps here (copy-and-paste from the PDF may give you line feeds you must remove first).

```
Enter your string.

But, soft! what light through yonder window breaks? It is the east, and Juliet
is the sun. Arise, fair sun, and kill the envious moon, Who is already sick and
pale with grief, That thou her maid art far more fair than she: Be not her maid,
since she is envious; Her vestal livery is but sick and green And none but fools
do wear it; cast it off. It is my lady, O, it is my love! O, that she knew she
were! She speaks yet she says nothing: what of that? Her eye discourses; I will
answer it. I am too bold, 'tis not to me she speaks: Two of the fairest stars in
all the heaven, Having some business, do entreat her eyes To twinkle in their
spheres till they return. What if her eyes were there, they in her head? The
brightness of her cheek would shame those stars, As daylight doth a lamp; her
eyes in heaven Would through the airy region stream so bright That birds would
sing and think it were not night. See, how she leans her cheek upon her hand! O,
that I were a glove upon that hand, That I might touch that cheek!
```

You entered: But, soft! what light through yonder window breaks? It is the east,
and Juliet is the sun. Arise, fair sun, and kill the envious moon, Who is
already sick and pale with grief, That thou her maid art far more fair than she:
Be not her maid, since she is envious; Her vestal livery is but sick and green
And none but fools do wear it; cast it off. It is my lady, O, it is my love! O,
that she knew she were! She speaks yet she says nothing: what of that? Her eye
discourses; I will answer it. I am too bold, 'tis not to me she speaks: Two of
the fairest stars in all the heaven, Having some business, do entreat her eyes
to twinkle in their spheres till they return. What if her eyes were there, they in
her head? The brightness of her cheek would shame those stars, As daylight
doth a lamp; her eyes in heaven Would through the airy region stream so bright
That birds would sing and think it were not night. See, how she leans her cheek
upon her hand! O, that I were a glove upon that hand, That I might touch that
cheek!

Character Count:
----------------
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
59 10  9 25103 13 15 72 60  1 12 26 13 46 49  7  0 47 65 80 20  9 21  0 16  0

### 3. Deliverables

Submit a zip file containing the items listed below to the CourSys: [https://courses.cs.sfu.ca/](https://courses.cs.sfu.ca/)

1. vectorStats.cpp
2. letterCount.cpp
3. Any additional required files (such as ANSI codes header)

To submit, create a group in CourSys (even if you are working alone). You can do this under this assignment's activity, look for the link in the top right corner. Only one member of your group submits the assignment to CourSys.

Please remember that all submissions will automatically be compared for unexplainable similarities. This comparison will also include similar assignments from previous semesters and programs on the internet. Please review the notes from lecture on the expectations for academic honesty.