CECS 275: LAB – MODE

REMEMBER: The homework tips are here, and the coding standards are here.

OBJECTIVE: Drive home that arrays and pointers share a great deal in common, and the use of reference variables.

BACKGROUND: In any given array of values, the mode is the value that occurs most frequently. If we have an integer array: {1, 5, 9, 5, 1, 3, 5, 10, 8} for instance, 1 occurs 2 times, 3 occurs once, 5 occurs 3 times, 8 occurs only once, as does 10. So 5 is the mode of this set of values. Where it gets more interesting is if there is a "tie" for the most frequent values. Consider {1, 2, 3, 3, 2, 1, 5, 8, 9}. 1 occurs twice, 2 does as well, 3 occurs twice as well, and 5, 8, and 9 all occur just once. So you have three different values that each occur the greatest number of time in this list of values.

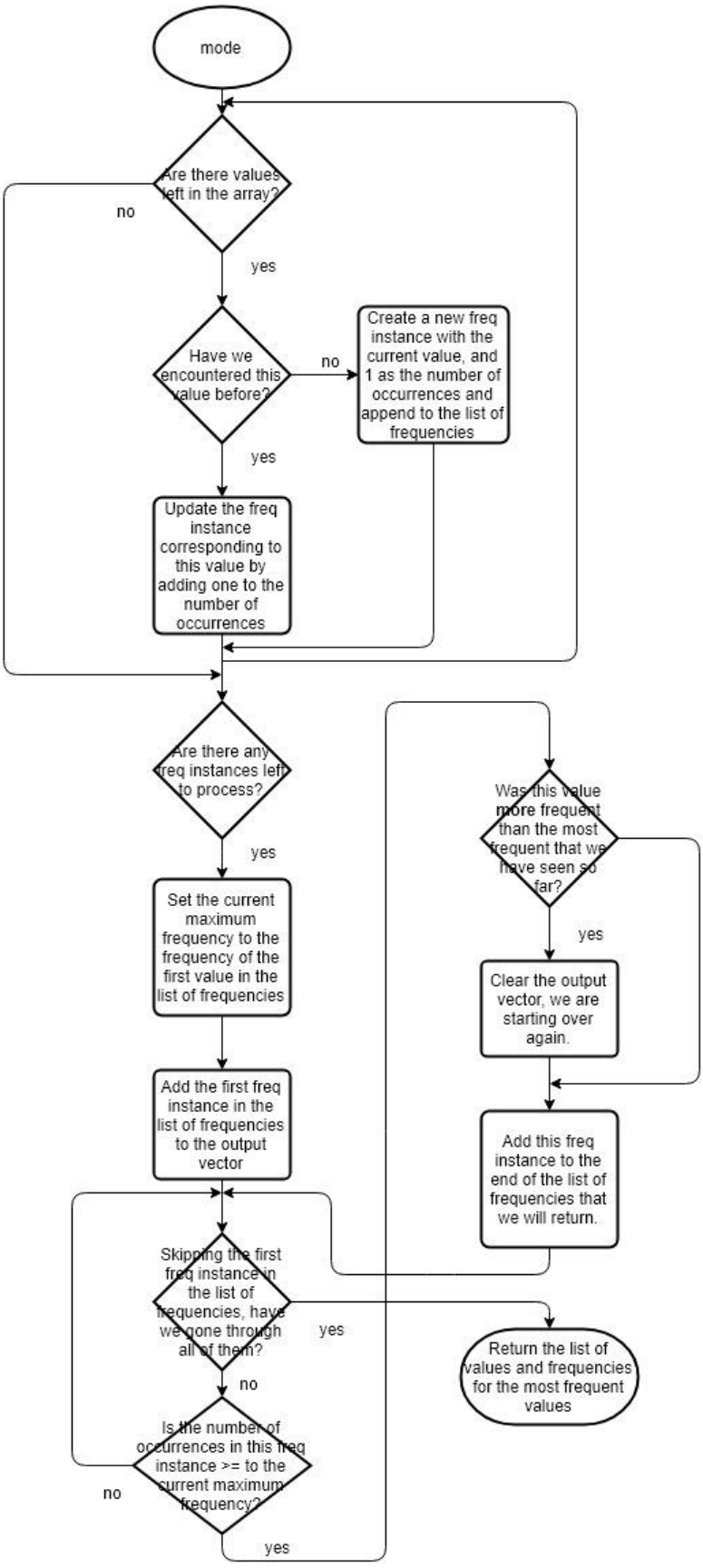
PROCEDURE:

- The overall flow of your application will be:
  - Prompt the user for the number of elements that they want in the list
    - Be sure that the user enters a valid integer, that is between 1 and the maximum number of array elements that you have room for. Do not use literal values (like the maximum number of array elements). Instead, assign the literal to a constant variable and use the variable when you need that value.
    - Remember that we are using a fixed-length array for this. Therefore, you want your array of integers to have some size larger than the user will ever-likely use. One hundred seems like a good compromise between too small to be useful and too large to fit into memory.
    - Then, you will have to keep track of how many elements the user actually enters each time, and pass that in to your mode function.
  - Go into a loop where you prompt the user for the values to put into the static array.
    - Make sure that the user enters valid integers. Remember that cin >> is all character-based, so you need to prompt the user again if they do not enter a valid integer.
  - Call your mode function (more on that in a moment) and report out the value(s) that occurred the most often in the array, and how often they occurred.
- Your mode function returns a vector of freq instances (more on those in a minute) that represent the value(s) that occurred most frequently in the sequence. The input arguments are:

  - array – The list of values that your user entered in the main function. Treat this strictly as a pointer inside your mode function. That means that you will need to use pointer arithmetic to go from one element in the array to the next.
  - nValues – The number of values in this particular array. Do not worry about initializing the array between calls to the mode function.
- A struct called freq that has just two member variables:
  - value – the value found in the list of values
  - frequency – the number of times that value showed up in the array
- findFreq – a function that locates a freq instance in a list of freqs that matches on the value. You will need this in your mode function. See the flowchart below.

Remember to validate all of your user input.

Flowchart for the mode function:


mode
Are there values left in the array?
no
yes
Have we encountered this value before?
no
Create a new freq instance with the current value, and 1 as the number of occurrences and append to the list of frequencies
yes
Update the freq instance corresponding to this value by adding one to the number of occurrences
Are there any freq instances left to process?
yes
Set the current maximum frequency to the frequency of the first value in the list of frequencies
Add the first freq instance in the list of frequencies to the output vector
Skipping the first freq instance in the list of frequencies, have we gone through all of them?
yes
no
Is the number of occurrences in this freq instance >= to the current maximum frequency?
no
yes
Was this value more frequent than the most frequent that we have seen so far?
yes
Clear the output vector, we are starting over again.
Add this freq instance to the end of the list of frequencies that we will return.
Return the list of values and frequencies for the most frequent values

Sample Output:

```
How many values --> 1
Enter the next value --> 5
The next value/frequency is: 5/1
Continue (y/n) --> y
How many values --> 10
Enter the next value --> 1
Enter the next value --> 1
Enter the next value --> 2
Enter the next value --> 2
Enter the next value --> 3
Enter the next value --> 3
Enter the next value --> 3
Enter the next value --> 4
Enter the next value --> 4
Enter the next value --> 4
The next value/frequency is: 3/3
The next value/frequency is: 4/3
Continue (y/n) --> y
How many values --> 10
Enter the next value --> 1
Enter the next value --> 1
Enter the next value --> 2
Enter the next value --> 2
Enter the next value --> 2
Enter the next value --> 3
Enter the next value --> 4
Enter the next value --> 4
Enter the next value --> 4
Enter the next value --> 5
The next value/frequency is: 2/3
The next value/frequency is: 4/3
Continue (y/n) --> y
How many values --> 5
Enter the next value --> 1
Enter the next value --> 2
Enter the next value --> 3
Enter the next value --> 4
Enter the next value --> 5
The next value/frequency is: 1/1
The next value/frequency is: 2/1
The next value/frequency is: 3/1
The next value/frequency is: 4/1
The next value/frequency is: 5/1
Continue (y/n) --> y
How many values --> 10
Enter the next value --> 1
Enter the next value --> 2
Enter the next value --> 3
Enter the next value --> 4
Enter the next value --> 5
Enter the next value --> 6
Enter the next value --> 7
```

```
Enter the next value --> 8
Enter the next value --> 9
Enter the next value --> 1
The next value/frequency is: 1/2
Continue (y/n) --> n
Completed satisfactorily
```

WHAT TO TURN IN for Lab Mode:

- Drop your source code file into the BeachBoard drop box.
- The filled out collaborate.doc file to let me know how/what each of you contributed to the overall result.
- Sample output in the file: output.txt.
- Do not forget to demonstrate your code to me before you leave.