

EE402 Tutorial 3 – More C and C++ with a bit of STL

Please solve the following short C++ problems. Your code should be concise and as efficient as possible, while still preserving best practice - i.e. avoid cutting-and-pasting any code. Note: **Each tutorial will be designed to have questions that become progressively more difficult, from fairly straightforward to very challenging.**

Q1 Write a C++ function of the form `int wordsCount(string[], int, int)`, where the first `int` is the size of the string array and the second `int` is the length (in characters) of the strings we wish to count. So, given an array of strings, return the count of the number of strings with the given length.

```
wordsCount({"a", "bb", "b", "ccc"}, 4, 1) → 2
wordsCount({"a", "bb", "b", "ccc"}, 4, 3) → 1
wordsCount({"a", "bb", "b", "ccc"}, 4, 4) → 0
```

Q2. Write a C++ class called `Statistics` that has two states, an array of 50 doubles and an `int arrayLength` state that will store the length of the array (assuming that there could be less than 50 elements in the double array). Add a display method that displays the contents of the array. You should be able to construct an object of your `Statistics` class by using:

```
double anArray[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
Statistics s(anArray, 10);
s.display();
```

and this will display:

```
Array [1 2 3 4 5 6 7 8 9 10 ]
```

Q3. Modify your class in Q2 to add an `average()`, `max()` and `min()` methods, which return the average of the array, the maximum and the minimum value in the array respectively.

Q4. Modify your class in Q2/Q3 so that it will accept an array of any length. Hint: you will have to dynamically allocate the space for the array using `new` and will have to use pointers. Ensure that all functions continue to work.

Note: Don't delete your `Statistics` class code just yet

Q5. Write a C++ function of the form `string mirrorEnds(string)`, that when given a string, looks for a mirror image (backwards) string at both the beginning and end of the given string. In other words, zero or more characters at the very beginning of the given string, and at the very end of the string in reverse order (possibly overlapping). For example, the string `"abXYZba"` has the mirror end `"ab"`.

```
mirrorEnds("abXYZba") → "ab"
mirrorEnds("abca") → "a"
```

```
mirrorEnds("navan") → "navan"
```

Q6. Using recursion only, write a C++ function of the form `int count7(int);` that when passed a non-negative `int n`, returns the number of occurrences of the 7 digit; so, for example 717 yields 2. (no loops!).

```
count7(717) → 2
count7(7) → 1
count7(123) → 0
```

Back to your Statistics class!

Q7. Comment out your `average()`, `min()` and `max()` methods. Change your `Statistics` class from Q4 so that it uses a **STL vector of doubles** to store the elements within the class. You should also remove the `arrayLength` state from your class as it should no longer be required.

Q8. Add back in (uncomment?) your `average()`, `min()` and `max()` methods to now use the `vector of double` state of the class.

Q9. Add `sort()` and `median()` methods to your statistics class that take advantage of the algorithms that are available through STL. The median is the centre value after sorting an array. However, if there is an even number of elements then it is the average of the pair of values at the centre.

Q10. There is an algorithm in STL called `random_shuffle`, which has the form:

```
void random_shuffle ( RandomAccessIterator first,
                    RandomAccessIterator last);
```

Use this algorithm to create a new method in your class called `randomize()`;

Also, there is another algorithm of the form:

```
int count = (int) count_if ( ForwardIterator first,
                          ForwardIterator last, Predicate pred);
```

that increments by 1 if the function `pred` returns true. Use this `count_if` algorithm to add two new methods to our class, `numberEvens()` and `numberOdds()` that returns a count of the number of (whole number part) even numbers and odd numbers respectively. Note: A suitable predicate function object is below:

```
class IsEvenPredicate {
public:
    bool operator () (double d)
    {
        //fill in the logic
    }
};
```