**Exercise 14**

**Create your own custom *String* class:**

**1.** The class can use the *std::string* variable within and should consist of a constructor that takes in a (optional) string argument.

**2.** The class should contain functions to add and remove pieces of string. The segments to add or remove should be passed in as string arguments.

**3.** Add a function called *Flush()* that will remove all characters from the string text.

**4.** Add a function called *Magnitude()* that returns the size of the string text.

**5.** Create a member function called *DisplayCharacters()* that will iterate through the entire string and display each and every character, all separated by commas. For example, the word *Hello* would display: **‘H’, ‘e’, ‘l’, ‘l’, and ‘o’**

**6.** Overload the **+** and **+=** operator that will add a passed string text to the current stored string. Equally, overload the **–** and **-=** operator to remove specific segments from the string.

**7.** Overload the **<** and **>** operator that will take in another *String* object and compare their sizes and return *true* or *false* accordingly.

**8.** Overload the **[]** operator such that an individual character from the string can be indexed and accessed so that characters can be acquired or assigned.

**9.** Add a function that checks whether the string text is a *palindrome* or not. A palindrome is a word or sentence that reads the same forward as it does backward.

**a.** Examples of palindrome words are:

*wow*, *kayak*, *rotator*, *civic*, *level*, *madam*, *racecar*, *tenet*.

**b.** Examples of palindrome phrases are:

*taco cat*, *my gym*, *red rum sir is murder*, *top spot*, *no lemon* *no melon*, *never odd or even*.

**c.** Examples of palindrome sentences are:

*Don't nod*, *I did did I*, *Step on no pets*, *Eva can I see bees in a cave*, *Was it a cat I saw*.