

```
----- Testing basic String creation & printing
string [0] = Wow
string [1] = C++ is neat!
string [2] =
string [3] = a-z

----- Testing access to characters (using const)
Whole string is abcdefghijklmnopqsrtuvwxyz
now char by char: abcdefghijklmnopqsrtuvwxyz
----- Testing access to characters (using non-const)
Start with abcdefghijklmnopqsrtuvwxyz and convert to ABCDEFGHIJKLMNOPQRSTUVWXYZ

----- Testing relational operators between MyStrings
Comparing app to apple
    Is left < right? true
    Is left <= right? true
    Is left > right? false
    Is left >= right? false
    Does left == right? false
    Does left != right ? true
Comparing apple to
    Is left < right? false
    Is left <= right? false
    Is left > right? true
    Is left >= right? true
    Does left == right? false
    Does left != right ? true
Comparing  to Banana
    Is left < right? true
    Is left <= right? true
    Is left > right? false
    Is left >= right? false
    Does left == right? false
    Does left != right ? true
Comparing Banana to Banana
    Is left < right? false
    Is left <= right? true
    Is left > right? false
    Is left >= right? true
    Does left == right? true
    Does left != right ? false

----- Testing relations between MyStrings and char *
Comparing he to hello
    Is left < right? true
    Is left <= right? true
    Is left > right? false
    Is left >= right? false
    Does left == right? false
    Does left != right ? true
Comparing why to wackity
    Is left < right? false
    Is left <= right? false
    Is left > right? true
    Is left >= right? true
    Does left == right? false
    Does left != right ? true

----- Testing copy constructor and operator= on MyStrings
original is cake, copy is fake
original is cake, copy is fake
after self assignment, copy is Copy Cat
Testing pass & return MyStrings by value and ref
```

```
after calling Append, sum is Binky  
val is winky  
after assign, val is Binky
```