Homework-09_cat_owl.txt

Names: Sean Hoerger, Marco Conati

HW Team: Homework-09_cat_owl

[completeness]
cpplint = (’4’, ’Excellent’)
repostatus = (’2’, ’Developing’)

[correctness]
testing results = (’3’, ’Good’)

appendTests.err

```
==> appendTests.err <==
appending size 36 to 1 passed!
appending size 36 to 2 passed!
appending size 36 to 12 passed!
appending size 36 to 13 passed!
appending size 36 to 24 passed!
appending size 36 to 25 passed!
appending size 36 to 36 passed!
appending size 36 to 37 passed!
appending size 36 to 6395 passed!
appending size 36 to 10000 passed!
appending size 37 to 1 passed!
appending size 37 to 2 passed!
appending size 37 to 12 passed!
appending size 37 to 13 passed!
appending size 37 to 24 passed!
appending size 37 to 25 passed!
appending size 37 to 36 passed!
appending size 37 to 37 passed!
appending size 37 to 6395 passed!
appending size 37 to 10000 passed!
appending size 6395 to 1 passed!
appending size 6395 to 2 passed!
appending size 6395 to 12 passed!
appending size 6395 to 13 passed!
appending size 6395 to 24 passed!
appending size 6395 to 25 passed!
appending size 6395 to 36 passed!
appending size 6395 to 37 passed!
appending size 6395 to 6395 passed!
appending size 6395 to 10000 passed!
appending size 10000 to 1 passed!
appending size 10000 to 2 passed!
appending size 10000 to 12 passed!
appending size 10000 to 13 passed!
appending size 10000 to 24 passed!
appending size 10000 to 25 passed!
appending size 10000 to 36 passed!
appending size 10000 to 37 passed!
appending size 10000 to 6395 passed!
appending size 10000 to 10000 passed!
self-append passed!
append utilization test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 100   ../../stringtest-hw9-solution.cpp:657:  testAppendSpecificSize(firstPair,
 secondPair)
0       / 1     ../../stringtest-hw9-solution.cpp:661:  testSelfAppend()
0       / 1     ../../stringtest-hw9-solution.cpp:662:  appendUtilizationTest()
```

copyConstructorTests.err

```
==> copyConstructorTests.err <==
copy constructor test for size1 passed!
copy constructor test for size2 passed!
copy constructor test for size12 passed!
copy constructor test for size13 passed!
copy constructor test for size24 passed!
copy constructor test for size25 passed!
copy constructor test for size36 passed!
copy constructor test for size37 passed!
copy constructor test for size6395 passed!
copy constructor test for size10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:368:  copyConstructorTest(testPair)
```

defaultConstructorTests.err

```
==> defaultConstructorTests.err <==
default constructor test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 1     ../../stringtest-hw9-solution.cpp:344:  defaultConstructorTest()
```

equalityTests.err

```
==> equalityTests.err <==
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:727:  testEquality(testingString, testi
ngString)
```

eraseTests.err

```
==> eraseTests.err <==
erase all middle, size 24 passed!
erase all back, size 24 passed!
erase all chunk boundary, size 24 passed!
erase all random location, size 24 passed!
erase all front, size 25 passed!
erase all middle, size 25 passed!
erase all back, size 25 passed!
erase all chunk boundary, size 25 passed!
erase all random location, size 25 passed!
erase all front, size 36 passed!
erase all middle, size 36 passed!
erase all back, size 36 passed!
erase all chunk boundary, size 36 passed!
erase all random location, size 36 passed!
erase all front, size 37 passed!
erase all middle, size 37 passed!
erase all back, size 37 passed!
erase all chunk boundary, size 37 passed!
erase all random location, size 37 passed!
erase all front, size 6395 passed!
erase all middle, size 6395 passed!
erase all back, size 6395 passed!
erase all chunk boundary, size 6395 passed!
erase all random location, size 6395 passed!
erase all front, size 10000 passed!
erase all middle, size 10000 passed!
erase all back, size 10000 passed!
erase all chunk boundary, size 10000 passed!
erase all random location, size 10000 passed!
erase from different places in long string passed!
FAILURE (after 181 passes): ../../stringtest-hw9-solution.cpp:204:      (testingIterator
== testingString.end()) == (controlIterator == controlString.end())

Summary of affirmation failures for erase utilization test
----
Fails   / Total Issue
2520    / 12200 ../../stringtest-hw9-solution.cpp:204:  (testingIterator == testingString
.end()) == (controlIterator == controlString.end())

FAILURE (after 0 passes): ../../stringtest-hw9-solution.cpp:1163:       eraseUtilizationT
est()

Summary of affirmation failures for All tests
----
Fails   / Total Issue
0       / 20    ../../stringtest-hw9-solution.cpp:1011: modifyIteratorTest(testPair, FRON
T, modificationType)
1       / 20    ../../stringtest-hw9-solution.cpp:1012: modifyIteratorTest(testPair, MIDD
LE, modificationType)
2       / 20    ../../stringtest-hw9-solution.cpp:1013: modifyIteratorTest(testPair, BACK
, modificationType)
0       / 16    ../../stringtest-hw9-solution.cpp:1015: modifyIteratorTest(testPair, CHUN
K, modificationType)
1       / 20    ../../stringtest-hw9-solution.cpp:1017: modifyIteratorTest(testPair, RAND
OM, modificationType)
0       / 1     ../../stringtest-hw9-solution.cpp:1162: alternatingErasesLongString()
1       / 1     ../../stringtest-hw9-solution.cpp:1163: eraseUtilizationTest()
```

inequalityTests.err

```
==> inequalityTests.err <==
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:708:  testInequality(testingString, new
TestingString)
0       / 10    ../../stringtest-hw9-solution.cpp:719:  testInequality(testingString, new
TestingString)
```

insertionTests.err

```
==> insertionTests.err <==
insert front, size 12 passed!
insert middle, size 12 passed!
insert back, size 12 passed!
insert chunk boundary, size 12 passed!
insert random location, size 12 passed!
insert front, size 13 passed!
insert middle, size 13 passed!
insert back, size 13 passed!
insert chunk boundary, size 13 passed!
insert random location, size 13 passed!
insert front, size 24 passed!
insert middle, size 24 passed!
insert back, size 24 passed!
insert chunk boundary, size 24 passed!
insert random location, size 24 passed!
insert front, size 25 passed!
insert middle, size 25 passed!
insert back, size 25 passed!
insert chunk boundary, size 25 passed!
insert random location, size 25 passed!
insert front, size 36 passed!
insert middle, size 36 passed!
insert back, size 36 passed!
insert chunk boundary, size 36 passed!
insert random location, size 36 passed!
insert front, size 37 passed!
insert middle, size 37 passed!
insert back, size 37 passed!
insert chunk boundary, size 37 passed!
insert random location, size 37 passed!
insert front, size 6395 passed!
insert middle, size 6395 passed!
insert back, size 6395 passed!
insert chunk boundary, size 6395 passed!
insert random location, size 6395 passed!
insert front, size 10000 passed!
insert middle, size 10000 passed!
insert back, size 10000 passed!
insert chunk boundary, size 10000 passed!
insert random location, size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:1011: modifyIteratorTest(testPair, FRON
T, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1012: modifyIteratorTest(testPair, MIDD
LE, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1013: modifyIteratorTest(testPair, BACK
, modificationType)
0       / 8     ../../stringtest-hw9-solution.cpp:1015: modifyIteratorTest(testPair, CHUN
K, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1017: modifyIteratorTest(testPair, RAND
OM, modificationType)
```

iteratorDefaultConstructorTests.err

```
==> iteratorDefaultConstructorTests.err <==
default iterator passed!
default const_iterator passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 1     ../../stringtest-hw9-solution.cpp:452:  defaultIteratorTest<false>()
0       / 1     ../../stringtest-hw9-solution.cpp:453:  defaultIteratorTest<true>()
```

iteratorTests.err

```
==> iteratorTests.err <==
iterator test for size 1 passed!
iterator test for size 2 passed!
iterator test for size 12 passed!
iterator test for size 13 passed!
iterator test for size 24 passed!
iterator test for size 25 passed!
iterator test for size 36 passed!
iterator test for size 37 passed!
iterator test for size 6395 passed!
iterator test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:526:  testIteratorSpecificSize(testPair
)
```

lessThanTests.err

```
==> lessThanTests.err <==
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:832:  testDefinitelyLessThan(empty, tes
tingString1)
0       / 10    ../../stringtest-hw9-solution.cpp:839:  testDefinitelyLessThan(testingStr
ing1, copy)
0       / 100   ../../stringtest-hw9-solution.cpp:854:  testLessThanUnknown(testingString
1, testingString2)
```

maketests.err

==> maketests.err <==

printTests.err

```
==> printTests.err <==
print test for size 1 passed!
print test for size 2 passed!
print test for size 12 passed!
print test for size 13 passed!
print test for size 24 passed!
print test for size 25 passed!
print test for size 36 passed!
print test for size 37 passed!
print test for size 6395 passed!
print test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:554:  testPrintSpecificSize(testPair)
```

sizeTests.err

```
==> sizeTests.err <==
size 1 test passed!
size 2 test passed!
size 12 test passed!
size 13 test passed!
size 24 test passed!
size 25 test passed!
size 36 test passed!
size 37 test passed!
size 6395 test passed!
size 10000 test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:394:  testSpecificSize(testPair)
```

utilizationTests.err

```
==> utilizationTests.err <==
utilization test for size 1 passed!
utilization test for size 2 passed!
utilization test for size 12 passed!
utilization test for size 13 passed!
utilization test for size 24 passed!
utilization test for size 25 passed!
utilization test for size 36 passed!
utilization test for size 37 passed!
utilization test for size 6395 passed!
utilization test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:418:  testUtilizationSpecificSize(testP
air)
```

weirdCharacterTests.err

```
==> weirdCharacterTests.err <==

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
```

appendTests.withvalgrind.err

```
==> appendTests.withvalgrind.err <==
appending size 24 to 1 passed!
appending size 24 to 2 passed!
appending size 24 to 12 passed!
appending size 24 to 13 passed!
appending size 24 to 24 passed!
appending size 24 to 25 passed!
appending size 24 to 36 passed!
appending size 24 to 37 passed!
appending size 24 to 6395 passed!
appending size 24 to 10000 passed!
appending size 25 to 1 passed!
appending size 25 to 2 passed!
appending size 25 to 12 passed!
appending size 25 to 13 passed!
appending size 25 to 24 passed!
appending size 25 to 25 passed!
appending size 25 to 36 passed!
appending size 25 to 37 passed!
appending size 25 to 6395 passed!
appending size 25 to 10000 passed!
appending size 36 to 1 passed!
appending size 36 to 2 passed!
appending size 36 to 12 passed!
appending size 36 to 13 passed!
appending size 36 to 24 passed!
appending size 36 to 25 passed!
appending size 36 to 36 passed!
appending size 36 to 37 passed!
appending size 36 to 6395 passed!
appending size 36 to 10000 passed!
appending size 37 to 1 passed!
appending size 37 to 2 passed!
appending size 37 to 12 passed!
appending size 37 to 13 passed!
appending size 37 to 24 passed!
appending size 37 to 25 passed!
appending size 37 to 36 passed!
appending size 37 to 37 passed!
appending size 37 to 6395 passed!
appending size 37 to 10000 passed!
appending size 6395 to 1 passed!
appending size 6395 to 2 passed!
appending size 6395 to 12 passed!
appending size 6395 to 13 passed!
appending size 6395 to 24 passed!
appending size 6395 to 25 passed!
appending size 6395 to 36 passed!
appending size 6395 to 37 passed!
appending size 6395 to 6395 passed!
Timeout occurred!
```

copyConstructorTests.withvalgrind.err

```
==> copyConstructorTests.withvalgrind.err <==
copy constructor test for size1 passed!
copy constructor test for size2 passed!
copy constructor test for size12 passed!
copy constructor test for size13 passed!
copy constructor test for size24 passed!
copy constructor test for size25 passed!
copy constructor test for size36 passed!
copy constructor test for size37 passed!
copy constructor test for size6395 passed!
copy constructor test for size10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:368:  copyConstructorTest(testPair)
```

defaultConstructorTests.withvalgrind.err

```
==> defaultConstructorTests.withvalgrind.err <==
default constructor test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 1     ../../stringtest-hw9-solution.cpp:344:  defaultConstructorTest()
```

equalityTests.withvalgrind.err

```
==> equalityTests.withvalgrind.err <==
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:727:  testEquality(testingString, testi
ngString)
```

equalityTestsWithInsertTests.withvalgrind.err

```
==> equalityTestsWithInsertTests.withvalgrind.err <==
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!
equality test with insert passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:782:  equalityTestWithInsert(testingStr
ing, testingString)
```

eraseTests.withvalgrind.err

```
==> eraseTests.withvalgrind.err <==
erase front, size 10000 passed!
erase middle, size 10000 passed!
FAILURE (after 0 passes): ../../stringtest-hw9-solution.cpp:991:        *testingIterator
== newChar

Summary of affirmation failures for erase back, size 10000
----
Fails   / Total Issue
1       / 1     ../../stringtest-hw9-solution.cpp:991:  *testingIterator == newChar

erase chunk boundary, size 10000 passed!
erase random location, size 10000 passed!
erase all front, size 1 passed!
erase all middle, size 1 passed!
erase all back, size 1 passed!
erase all random location, size 1 passed!
erase all front, size 2 passed!
erase all middle, size 2 passed!
erase all back, size 2 passed!
erase all random location, size 2 passed!
erase all front, size 12 passed!
erase all middle, size 12 passed!
erase all back, size 12 passed!
erase all chunk boundary, size 12 passed!
erase all random location, size 12 passed!
erase all front, size 13 passed!
erase all middle, size 13 passed!
erase all back, size 13 passed!
erase all chunk boundary, size 13 passed!
erase all random location, size 13 passed!
erase all front, size 24 passed!
erase all middle, size 24 passed!
erase all back, size 24 passed!
erase all chunk boundary, size 24 passed!
erase all random location, size 24 passed!
erase all front, size 25 passed!
erase all middle, size 25 passed!
erase all back, size 25 passed!
erase all chunk boundary, size 25 passed!
erase all random location, size 25 passed!
erase all front, size 36 passed!
erase all middle, size 36 passed!
erase all back, size 36 passed!
erase all chunk boundary, size 36 passed!
erase all random location, size 36 passed!
erase all front, size 37 passed!
erase all middle, size 37 passed!
erase all back, size 37 passed!
erase all chunk boundary, size 37 passed!
erase all random location, size 37 passed!
Timeout occurred!
```

inequalityTests.withvalgrind.err

```
==> inequalityTests.withvalgrind.err <==
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!
inequality test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:708:  testInequality(testingString, new
TestingString)
0       / 10    ../../stringtest-hw9-solution.cpp:719:  testInequality(testingString, new
TestingString)
```

insertionTests.withvalgrind.err

```
==> insertionTests.withvalgrind.err <==
insert front, size 12 passed!
insert middle, size 12 passed!
insert back, size 12 passed!
insert chunk boundary, size 12 passed!
insert random location, size 12 passed!
insert front, size 13 passed!
insert middle, size 13 passed!
insert back, size 13 passed!
insert chunk boundary, size 13 passed!
insert random location, size 13 passed!
insert front, size 24 passed!
insert middle, size 24 passed!
insert back, size 24 passed!
insert chunk boundary, size 24 passed!
insert random location, size 24 passed!
insert front, size 25 passed!
insert middle, size 25 passed!
insert back, size 25 passed!
insert chunk boundary, size 25 passed!
insert random location, size 25 passed!
insert front, size 36 passed!
insert middle, size 36 passed!
insert back, size 36 passed!
insert chunk boundary, size 36 passed!
insert random location, size 36 passed!
insert front, size 37 passed!
insert middle, size 37 passed!
insert back, size 37 passed!
insert chunk boundary, size 37 passed!
insert random location, size 37 passed!
insert front, size 6395 passed!
insert middle, size 6395 passed!
insert back, size 6395 passed!
insert chunk boundary, size 6395 passed!
insert random location, size 6395 passed!
insert front, size 10000 passed!
insert middle, size 10000 passed!
insert back, size 10000 passed!
insert chunk boundary, size 10000 passed!
insert random location, size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:1011: modifyIteratorTest(testPair, FRON
T, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1012: modifyIteratorTest(testPair, MIDD
LE, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1013: modifyIteratorTest(testPair, BACK
, modificationType)
0       / 8     ../../stringtest-hw9-solution.cpp:1015: modifyIteratorTest(testPair, CHUN
K, modificationType)
0       / 10    ../../stringtest-hw9-solution.cpp:1017: modifyIteratorTest(testPair, RAND
OM, modificationType)
```

iteratorDefaultConstructorTests.withvalgrind.er

```
==> iteratorDefaultConstructorTests.withvalgrind.err <==
default iterator passed!
default const_iterator passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 1     ../../stringtest-hw9-solution.cpp:452:  defaultIteratorTest<false>()
0       / 1     ../../stringtest-hw9-solution.cpp:453:  defaultIteratorTest<true>()
```

iteratorTests.withvalgrind.err

```
==> iteratorTests.withvalgrind.err <==
iterator test for size 1 passed!
iterator test for size 2 passed!
iterator test for size 12 passed!
iterator test for size 13 passed!
iterator test for size 24 passed!
iterator test for size 25 passed!
iterator test for size 36 passed!
iterator test for size 37 passed!
iterator test for size 6395 passed!
iterator test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:526:  testIteratorSpecificSize(testPair
)
```

lessThanTests.withvalgrind.err

```
==> lessThanTests.withvalgrind.err <==
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
definitely less-than passed!
definitely less-than passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!
less-than unknown passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:832:  testDefinitelyLessThan(empty, tes
tingString1)
0       / 10    ../../stringtest-hw9-solution.cpp:839:  testDefinitelyLessThan(testingStr
ing1, copy)
0       / 100   ../../stringtest-hw9-solution.cpp:854:  testLessThanUnknown(testingString
1, testingString2)
```

printTests.withvalgrind.err

```
==> printTests.withvalgrind.err <==
print test for size 1 passed!
print test for size 2 passed!
print test for size 12 passed!
print test for size 13 passed!
print test for size 24 passed!
print test for size 25 passed!
print test for size 36 passed!
print test for size 37 passed!
print test for size 6395 passed!
print test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:554:  testPrintSpecificSize(testPair)
```

sizeTests.withvalgrind.err

```
==> sizeTests.withvalgrind.err <==
size 1 test passed!
size 2 test passed!
size 12 test passed!
size 13 test passed!
size 24 test passed!
size 25 test passed!
size 36 test passed!
size 37 test passed!
size 6395 test passed!
size 10000 test passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:394:  testSpecificSize(testPair)
```

utilizationTests.withvalgrind.err

```
==> utilizationTests.withvalgrind.err <==
utilization test for size 1 passed!
utilization test for size 2 passed!
utilization test for size 12 passed!
utilization test for size 13 passed!
utilization test for size 24 passed!
utilization test for size 25 passed!
utilization test for size 36 passed!
utilization test for size 37 passed!
utilization test for size 6395 passed!
utilization test for size 10000 passed!

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
0       / 10    ../../stringtest-hw9-solution.cpp:418:  testUtilizationSpecificSize(testP
air)
```

weirdCharacterTests.withvalgrind.err

```
==> weirdCharacterTests.withvalgrind.err <==

All tests passed! Summary of affirmations:
----
Fails   / Total Issue
```

Insert Source Code and Helpers

```
161  // 1 Param Chunk Constructor
162  ChunkyString::Chunk::Chunk(size_t length) : length_{length} {
163      // nothing left to do
164  }

166  /*
167    Insert a character and return the index of the insertee. If the insertion
168    is at the end of a chunk, just push back. Otherwise, if the chunk is full
169    split the chunk into the current chunk and a new chunk. Push half the
170    elements to the new chunk and fix length.
171    In any case, at this point, perform a regular insert. To do so,
172    move all elements after the index down one and insert the insertee
173    into the correct index. Fix size and return the iterator.
174  */
175  ChunkyString::Iterator ChunkyString::insert(ChunkyString::Iterator i, char c) {
176      // if the insertion is at the end, this should be equivalent to a push back
177      if (i == end()) {
178          push_back(c);
179          iterator toReturn = end();
180          --toReturn;
181          return toReturn;
182      }
183      // If the chunk is full, make a new chunk
184      if (i.chunkPointer_->length_ >= (Chunk::CHUNKSIZE)) {
185          std::list<Chunk>::iterator nextChunk = i.chunkPointer_;
186          ++nextChunk;
187          Chunk newChunk = Chunk{0};
188          nextChunk = chunks_.insert(nextChunk, newChunk);
189
190          // push half the elements onto the new chunk and fix lengths
191          for (size_t n = 0; n < Chunk::CHUNKSIZE/2; ++n) {
192              nextChunk->chars_[n] =
193              i.chunkPointer_->chars_[Chunk::CHUNKSIZE/2 + n];
194              --(i.chunkPointer_->length_);
195              ++(nextChunk->length_);
196          }
197          // fix the iterator if the index is now pointing to a different chunk
198          if (i.index_ > Chunk::CHUNKSIZE/2) {
199              i = iterator{nextChunk, i.index_ - Chunk::CHUNKSIZE/2};
200          }
201      }
202
203      // now we have chunks that aren't full, so we can perform a regular insert
204      for (size_t n = i.chunkPointer_->length_; n > i.index_; --n) {
205          i.chunkPointer_->chars_[n] = i.chunkPointer_->chars_[n - 1];
206      }
207      // insert the new character
208      i.chunkPointer_->chars_[i.index_] = c;
209      // adjust length, size and return
210      ++(i.chunkPointer_->length_);
211      ++size_;
212      return i;
213  }
```

Erase Source Code and Helpers

```
214  /*
215      Erases the character at the iterator and returns the index of the next char
216
217      Case 1: If the chunk has one element, simply delete the chunk.
218      Case 2: If the erasee is last in a chunk, delete it and change the length
219      Case 3: If the character is in the beginning or the middle of the a chunk,
220              bump all of the elemtents up 1 and decrement length
221
222      Finally, for cases 2 and 3 call reorder to optimize utlization.
223  */
224  ChunkyString::Iterator ChunkyString::erase(ChunkyString::Iterator i) {
225      // if the chunk has one element, delete the chunk!
226      if (i.chunkPointer_->length_ == 1) {
227          --size_;
228          std::list<Chunk>::iterator deletedChunk =
229          chunks_.erase(i.chunkPointer_);
230          // return an iterator to the place of the deleted chunk
231          return iterator{deletedChunk, 0};
232      // otherwise, if the erasee is last in a chunk, delete it and change length
233      } else {
234          if (i.index_ == (i.chunkPointer_->length_ - 1)) {
235              --(i.chunkPointer_->length_);
236              i = Iterator{i.chunkPointer_, i.index_};
237          // finally, if the character is at the beginning or in the middle of
238          // a chunk, bump all other elements up 1 and decrement length
239          } else {
240              for (size_t n = i.index_; n < i.chunkPointer_->length_- 1; ++n) {
241                  i.chunkPointer_->chars_[n] = i.chunkPointer_->chars_[n + 1];
242              }
243              --(i.chunkPointer_->length_);
244          }
245          // decrement size
246          --size_;
247          // call this function to keep utilization low
248          i = reorder(i);
249      }
250      return i;
251  }
```