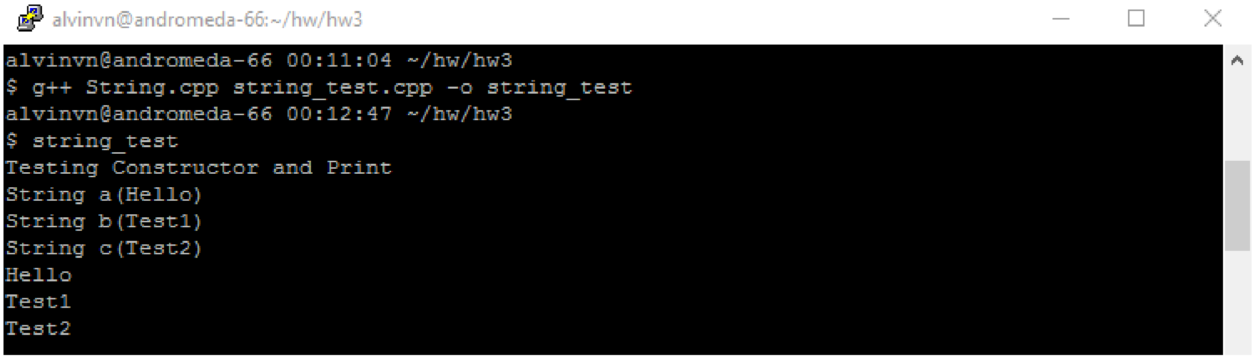
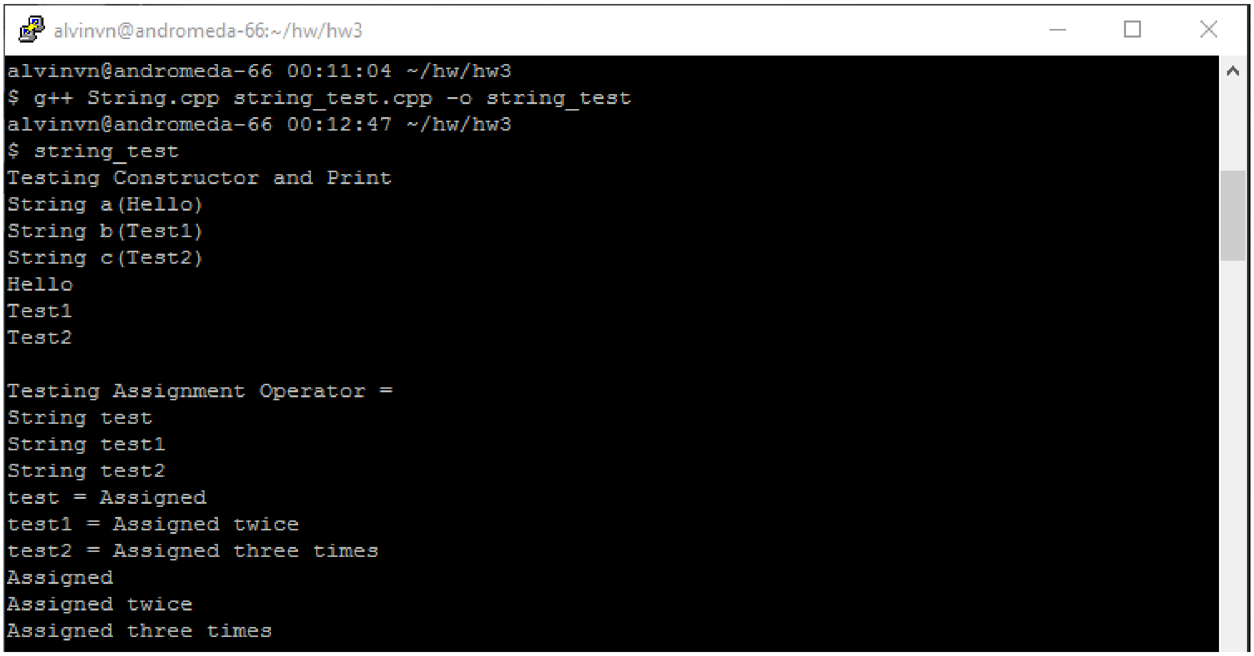
Testing Constructor and Print:



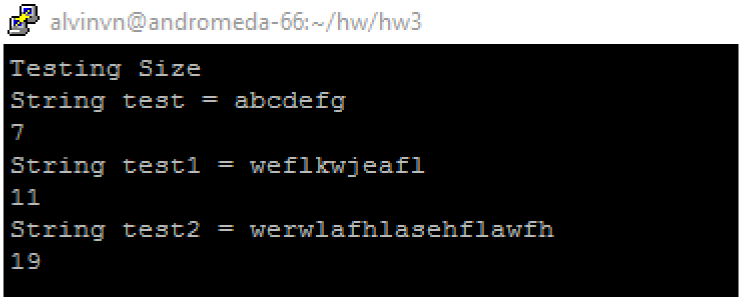
```
alvinvn@andromeda-66 00:11:04 ~/hw/hw3
$ g++ String.cpp string_test.cpp -o string_test
alvinvn@andromeda-66 00:12:47 ~/hw/hw3
$ string_test
Testing Constructor and Print
String a(Hello)
String b(Test1)
String c(Test2)
Hello
Test1
Test2
```

Testing Assignment Operator = :



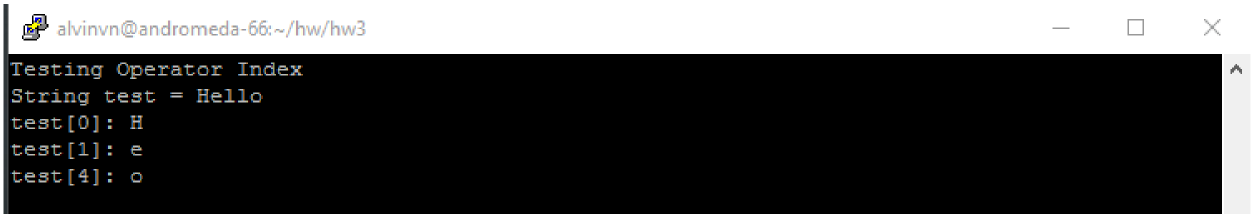
```
alvinvn@andromeda-66 00:11:04 ~/hw/hw3
$ g++ String.cpp string_test.cpp -o string_test
alvinvn@andromeda-66 00:12:47 ~/hw/hw3
$ string_test
Testing Constructor and Print
String a(Hello)
String b(Test1)
String c(Test2)
Hello
Test1
Test2

Testing Assignment Operator =
String test
String test1
String test2
test = Assigned
test1 = Assigned twice
test2 = Assigned three times
Assigned
Assigned twice
Assigned three times
```

Testing Size:



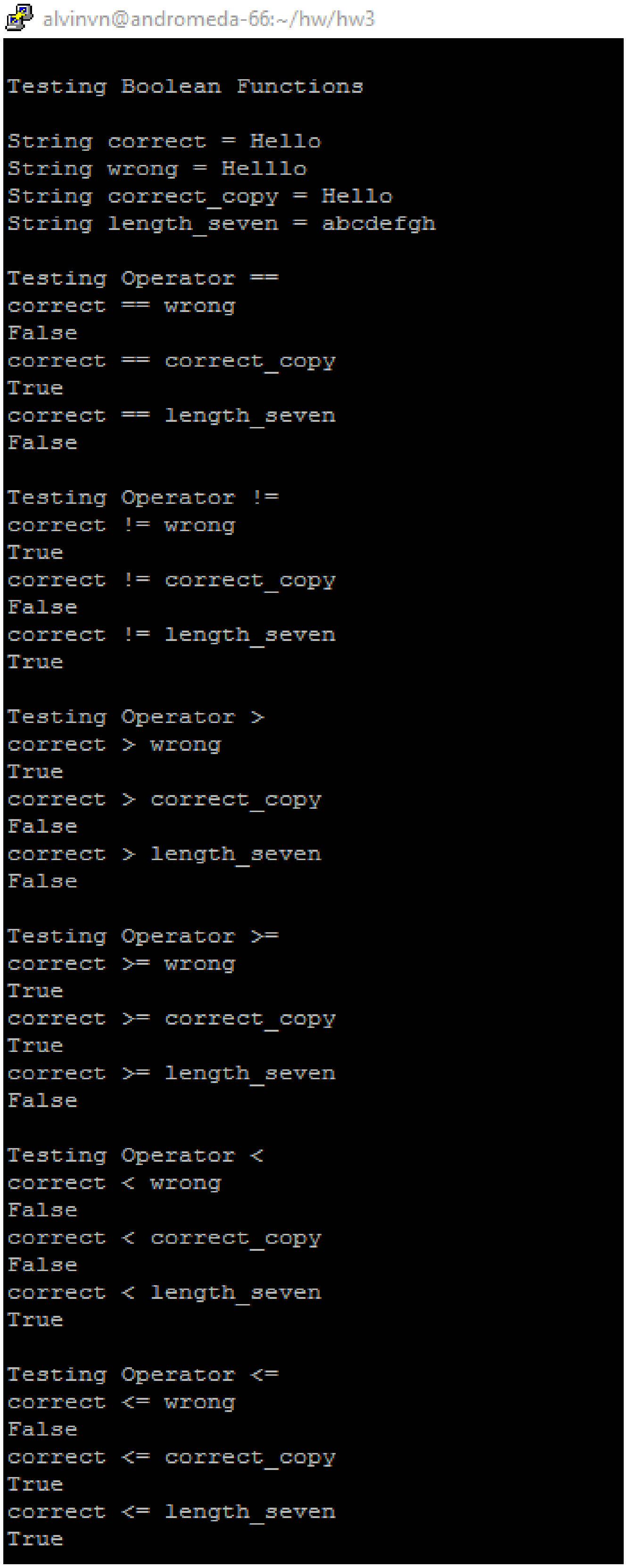
```
Testing Size
String test = abcdefg
7
String test1 = weflkwjeafl
11
String test2 = werwlafhlasehflawfh
19
```

Testing Operator Index:



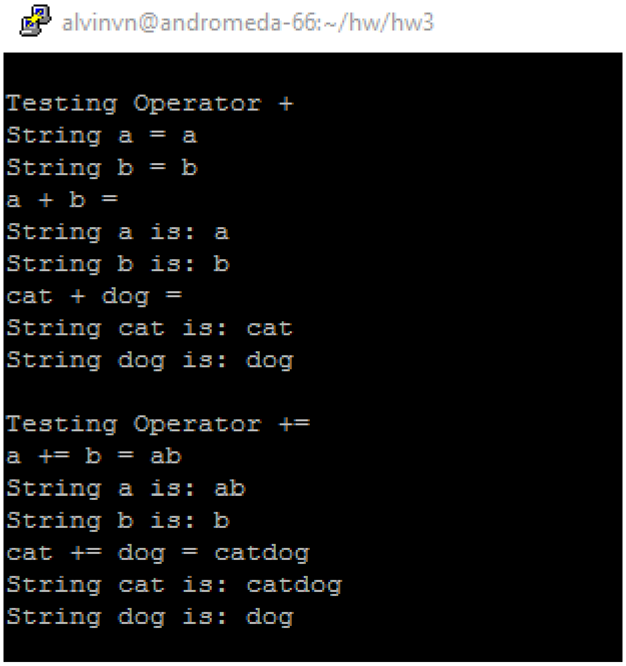
```
Testing Operator Index
String test = Hello
test[0]: H
test[1]: e
test[4]: o
```

Testing Boolean Functions (==, !=, > , < , <= , >=):



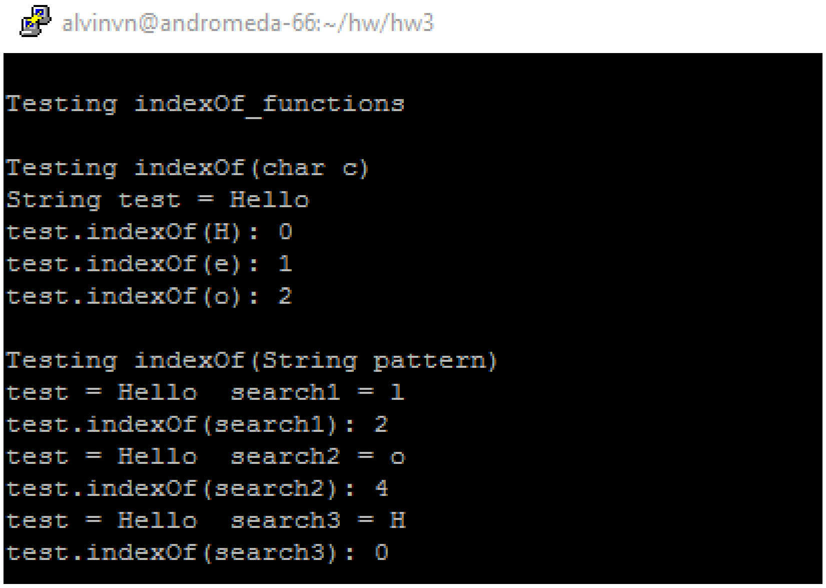
```
Testing Boolean Functions

String correct = Hello
String wrong = Helllo
String correct_copy = Hello
String length_seven = abcdefgh

Testing Operator ==
correct == wrong
False
correct == correct_copy
True
correct == length_seven
False

Testing Operator !=
correct != wrong
True
correct != correct_copy
False
correct != length_seven
True

Testing Operator >
correct > wrong
True
correct > correct_copy
False
correct > length_seven
False

Testing Operator >=
correct >= wrong
True
correct >= correct_copy
True
correct >= length_seven
False

Testing Operator <
correct < wrong
False
correct < correct_copy
False
correct < length_seven
True

Testing Operator <=
correct <= wrong
False
correct <= correct_copy
True
correct <= length_seven
True
```

Testing Operators + and +=:


alvinvn@andromeda-66:~/hw/hw3

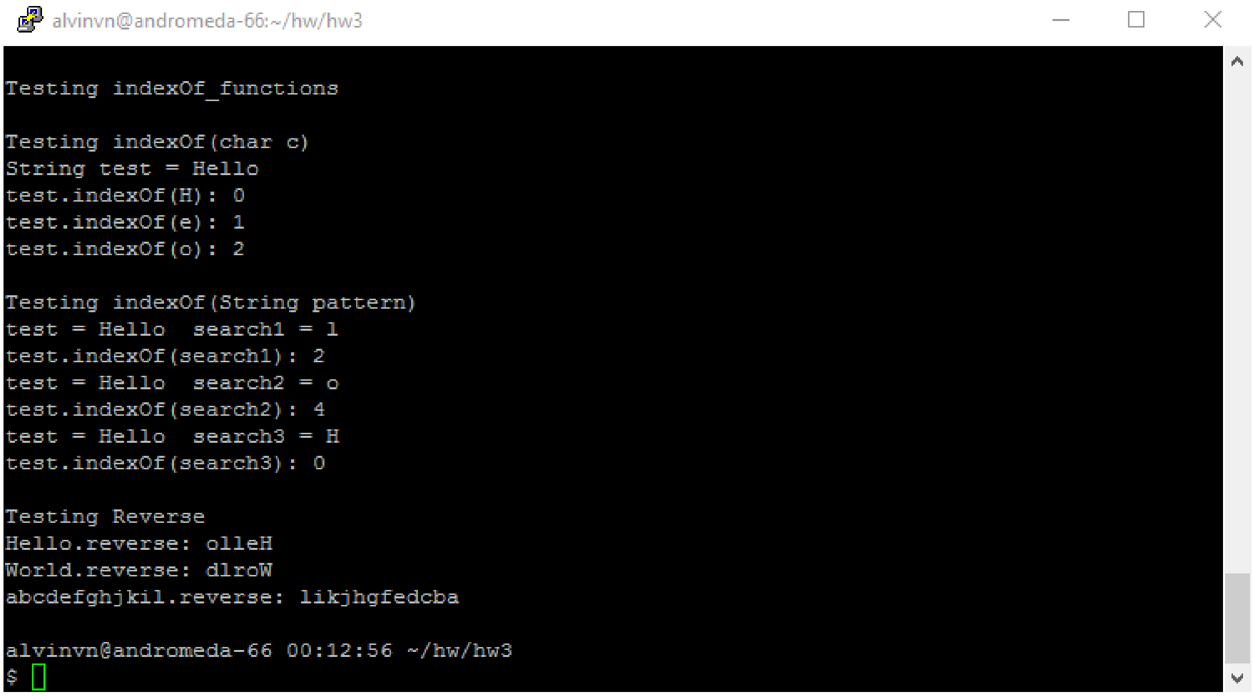
```
Testing Operator +
String a = a
String b = b
a + b =
String a is: a
String b is: b
cat + dog =
String cat is: cat
String dog is: dog

Testing Operator +=
a += b = ab
String a is: ab
String b is: b
cat += dog = catdog
String cat is: catdog
String dog is: dog
```

Testing Indexof Functions:


alvinvn@andromeda-66:~/hw/hw3

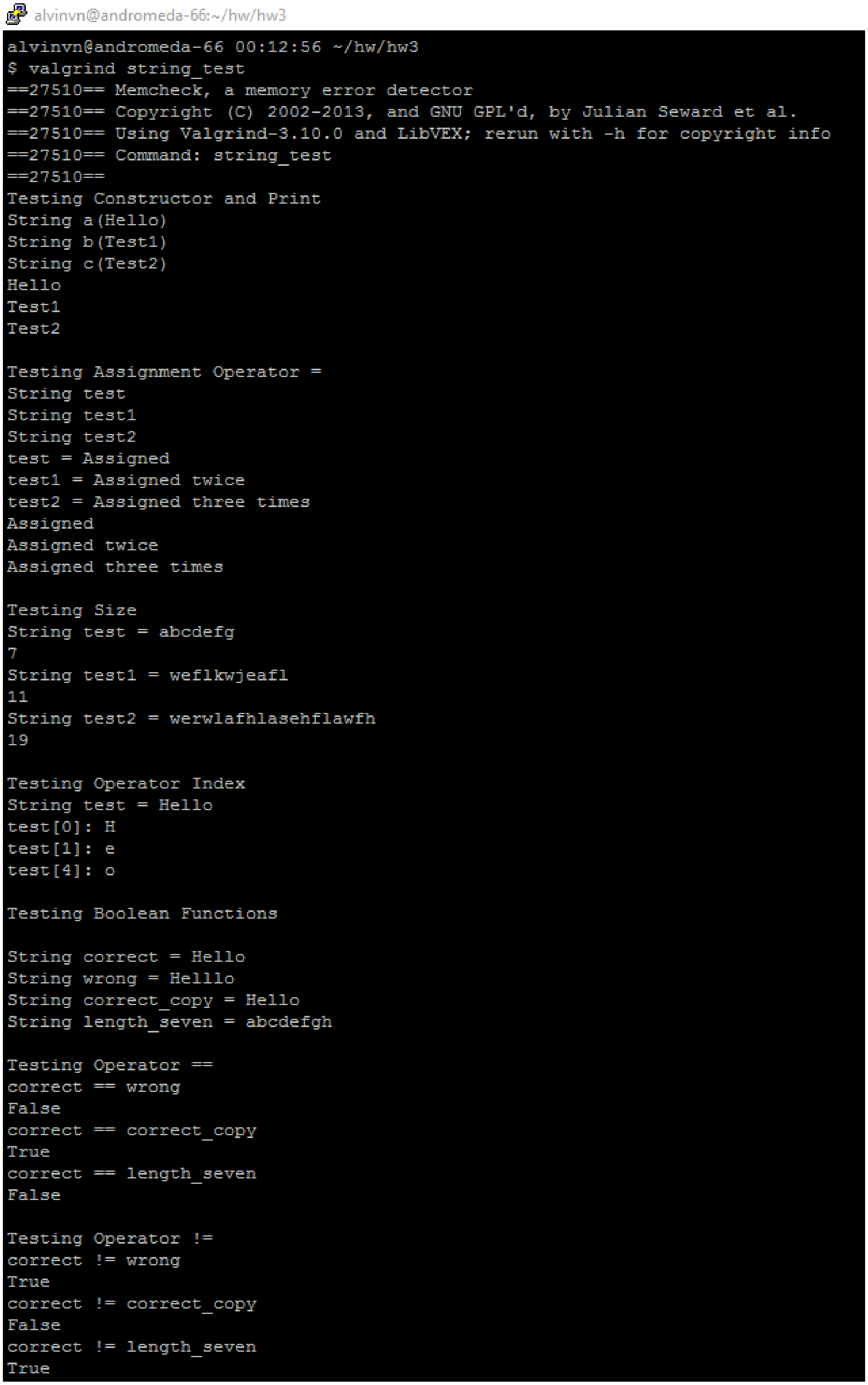
```
Testing indexOf_functions

Testing indexOf(char c)
String test = Hello
test.indexOf(H): 0
test.indexOf(e): 1
test.indexOf(o): 2

Testing indexOf(String pattern)
test = Hello  search1 = l
test.indexOf(search1): 2
test = Hello  search2 = o
test.indexOf(search2): 4
test = Hello  search3 = H
test.indexOf(search3): 0
```

Testing Reverse:

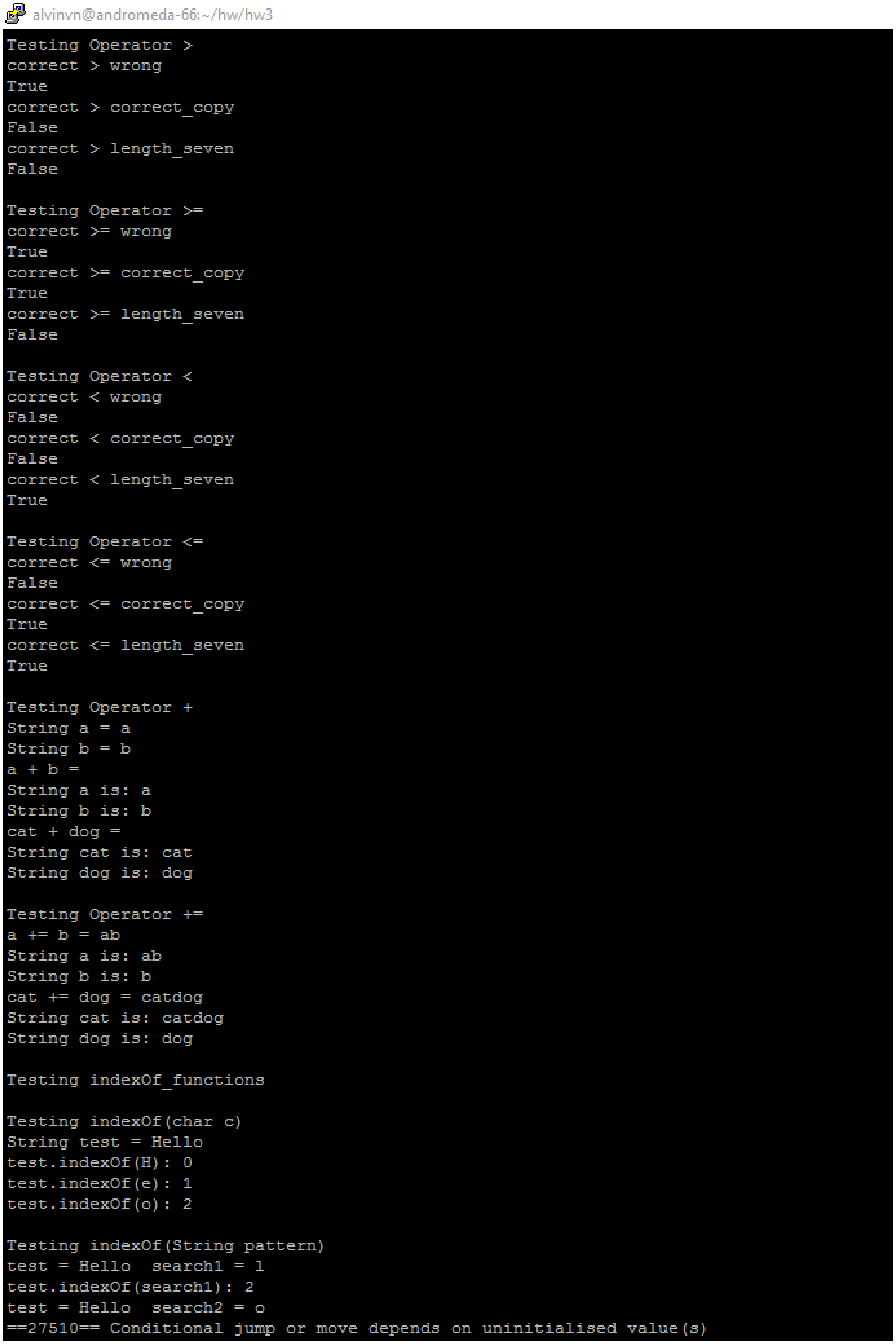


```
Testing indexOf_functions

Testing indexOf(char c)
String test = Hello
test.indexOf(H): 0
test.indexOf(e): 1
test.indexOf(o): 2

Testing indexOf(String pattern)
test = Hello  search1 = l
test.indexOf(search1): 2
test = Hello  search2 = o
test.indexOf(search2): 4
test = Hello  search3 = H
test.indexOf(search3): 0

Testing Reverse
Hello.reverse: olleH
World.reverse: dlroW
abcdefghjkil.reverse: likjhgfedcba

alvinvn@andromeda-66 00:12:56 ~/hw/hw3
$ 
```

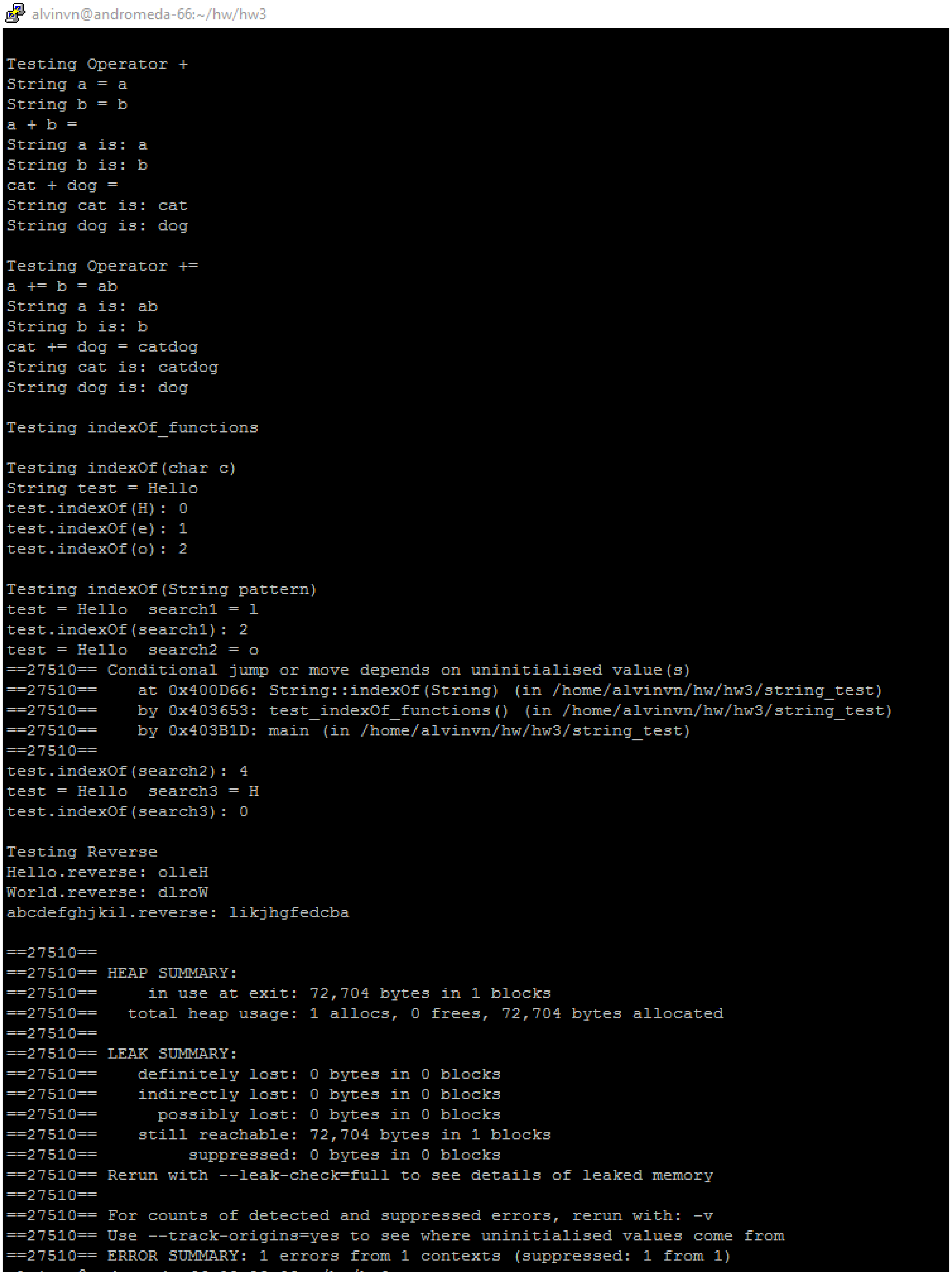
VALGRIND REPORT:



```
alvinvn@andromeda-66 00:12:56 ~/hw/hw3
$ valgrind string_test
==27510== Memcheck, a memory error detector
==27510== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==27510== Using Valgrind-3.10.0 and LibVEX; rerun with -h for copyright info
==27510== Command: string_test
==27510==
Testing Constructor and Print
String a(Hello)
String b(Test1)
String c(Test2)
Hello
Test1
Test2

Testing Assignment Operator =
String test
String test1
String test2
test = Assigned
test1 = Assigned twice
test2 = Assigned three times
Assigned
Assigned twice
Assigned three times

Testing Size
String test = abcdefg
7
String test1 = weflkwjeafl
11
String test2 = werwlafhlasehflawfh
19

Testing Operator Index
String test = Hello
test[0]: H
test[1]: e
test[4]: o

Testing Boolean Functions

String correct = Hello
String wrong = Helllo
String correct_copy = Hello
String length_seven = abcdefgh

Testing Operator ==
correct == wrong
False
correct == correct_copy
True
correct == length_seven
False

Testing Operator !=
correct != wrong
True
correct != correct_copy
False
correct != length_seven
True
```



```
Testing Operator >
correct > wrong
True
correct > correct_copy
False
correct > length_seven
False

Testing Operator >=
correct >= wrong
True
correct >= correct_copy
True
correct >= length_seven
False

Testing Operator <
correct < wrong
False
correct < correct_copy
False
correct < length_seven
True

Testing Operator <=
correct <= wrong
False
correct <= correct_copy
True
correct <= length_seven
True

Testing Operator +
String a = a
String b = b
a + b =
String a is: a
String b is: b
cat + dog =
String cat is: cat
String dog is: dog

Testing Operator +=
a += b = ab
String a is: ab
String b is: b
cat += dog = catdog
String cat is: catdog
String dog is: dog

Testing indexOf_functions

Testing indexOf(char c)
String test = Hello
test.indexOf(H): 0
test.indexOf(e): 1
test.indexOf(o): 2

Testing indexOf(String pattern)
test = Hello  search1 = l
test.indexOf(search1): 2
test = Hello  search2 = o
==27510== Conditional jump or move depends on uninitialised value(s)
```



```
Testing Operator +
String a = a
String b = b
a + b =
String a is: a
String b is: b
cat + dog =
String cat is: cat
String dog is: dog

Testing Operator +=
a += b = ab
String a is: ab
String b is: b
cat += dog = catdog
String cat is: catdog
String dog is: dog

Testing indexOf_functions

Testing indexOf(char c)
String test = Hello
test.indexOf(H): 0
test.indexOf(e): 1
test.indexOf(o): 2

Testing indexOf(String pattern)
test = Hello  search1 = l
test.indexOf(search1): 2
test = Hello  search2 = o
==27510== Conditional jump or move depends on uninitialised value(s)
==27510==    at 0x400D66: String::indexOf(String) (in /home/alvinvn/hw/hw3/string_test)
==27510==    by 0x403653: test_indexOf_functions() (in /home/alvinvn/hw/hw3/string_test)
==27510==    by 0x403B1D: main (in /home/alvinvn/hw/hw3/string_test)
==27510==
test.indexOf(search2): 4
test = Hello  search3 = H
test.indexOf(search3): 0

Testing Reverse
Hello.reverse: olleH
World.reverse: dlroW
abcdefghjkil.reverse: likjhgfedcba

==27510==
==27510== HEAP SUMMARY:
==27510==     in use at exit: 72,704 bytes in 1 blocks
==27510==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==27510==
==27510== LEAK SUMMARY:
==27510==    definitely lost: 0 bytes in 0 blocks
==27510==    indirectly lost: 0 bytes in 0 blocks
==27510==      possibly lost: 0 bytes in 0 blocks
==27510==    still reachable: 72,704 bytes in 1 blocks
==27510==         suppressed: 0 bytes in 0 blocks
==27510== Rerun with --leak-check=full to see details of leaked memory
==27510==
==27510== For counts of detected and suppressed errors, rerun with: -v
==27510== Use --track-origins=yes to see where uninitialised values come from
==27510== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 1 from 1)
```