MOHAMED KHARAEV 43121144

```
mkharaev@andromeda-8 01:50:31 ~/hw/hw4
$ g++ test_string.cpp
mkharaev@andromeda-8 01:50:47 ~/hw/hw4
$ a.out
Hello World
Test has a length of 4
The character at index 3 of Index is e
Assignment Test
Relations Tests
relations > test:  false
abra < kadabra: true
abra == abra: true
abra != kadabra: true
relations >= kadabra: true
test <= test: true
First String: This is string 1.
concatenation: This is string 1.This is string 2.
First String: This is string 1.
concatenation (+=): This is string 1.This is string 2.
First String: This is string 1.This is string 2.
Reverse "Four": ruoF
The index of 'o' in the string "Lorun" is: 1
The index of "um" in the string "Lorum" is: 3
Enter a test string: qwer
Your string: qwer
```

```
mkharaev@andromeda-8 01:51:54 ~/hw/hw4
$ g++ test_string.cpp
mkharaev@andromeda-8 01:52:08 ~/hw/hw4
$ valgrind a.out
==13220== Memcheck, a memory error detector
==13220== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==13220== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==13220== Command: a.out
==13220==
Hello World
Test has a length of 4
The character at index 3 of Index is e
Assignment Test
Relations Tests
relations > test:  false
abra < kadabra: true
abra == abra: true
abra != kadabra: true
relations >= kadabra: true
test <= test: true
First String: This is string 1.
concatenation: This is string 1.This is string 2.
First String: This is string 1.
concatenation (+=): This is string 1.This is string 2.
First String: This is string 1.This is string 2.
Reverse "Four": ruoF
The index of 'o' in the string "Lorun" is: 1
The index of "um" in the string "Lorum" is: 3
Enter a test string: qwer
Your string: qwer
==13220==
==13220== HEAP SUMMARY:
==13220==     in use at exit: 72,704 bytes in 1 blocks
==13220==   total heap usage: 40 allocs, 39 frees, 73,426 bytes allocated
==13220==
==13220== LEAK SUMMARY:
==13220==    definitely lost: 0 bytes in 0 blocks
==13220==    indirectly lost: 0 bytes in 0 blocks
==13220==      possibly lost: 0 bytes in 0 blocks
==13220==    still reachable: 72,704 bytes in 1 blocks
==13220==         suppressed: 0 bytes in 0 blocks
==13220== Rerun with --leak-check=full to see details of leaked memory
==13220==
==13220== For counts of detected and suppressed errors, rerun with: -v
==13220== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
mkharaev@andromeda-8 01:52:14 ~/hw/hw4
```

```
mkharaev@andromeda-8 01:52:38 ~/hw/hw4
$ g++ standard_main.cpp
mkharaev@andromeda-8 01:52:50 ~/hw/hw4
$ a.out
+: FirstSecond
+=: FirstSecond
indexOf(String): 5
indexOf(char): 4
LT: 0
GT: 1
LE: 0
GE: 1
<<:
<<: Fourth
==: 0
indexOf(String): 6
size(): 0
size(): 6
[]: i
reverse(): htruoF
<<: First First
[]: i
String index out of bounds
[]: F
!=: 0
Enter a test string: hello
hello
0
0
1
1
Number of new allocations minus number of delete deallocations is 0
```

```
mkharaev@andromeda-8 01:53:14 ~/hw/hw4
$ g++ standard_main.cpp
mkharaev@andromeda-8 01:53:22 ~/hw/hw4
$ valgrind a.out
==13239== Memcheck, a memory error detector
==13239== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==13239== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==13239== Command: a.out
==13239==
+: FirstSecond
+=: FirstSecond
indexOf(String): 5
indexOf(char): 4
LT: 0
GT: 1
LE: 0
GE: 1
<<:
<<: Fourth
==: 0
indexOf(String): 6
size(): 0
size(): 6
[]: i
reverse(): htruoF
<<: First First
[]: i
String index out of bounds
[]: F
!=: 0
Enter a test string: hello
hello
0
0
1
1
Number of new allocations minus number of delete deallocations is 0
==13239==
==13239== HEAP SUMMARY:
==13239==     in use at exit: 72,704 bytes in 1 blocks
==13239==   total heap usage: 35 allocs, 34 frees, 73,202 bytes allocated
==13239==
==13239== LEAK SUMMARY:
==13239==    definitely lost: 0 bytes in 0 blocks
==13239==    indirectly lost: 0 bytes in 0 blocks
==13239==      possibly lost: 0 bytes in 0 blocks
==13239==    still reachable: 72,704 bytes in 1 blocks
==13239==         suppressed: 0 bytes in 0 blocks
==13239== Rerun with --leak-check=full to see details of leaked memory
==13239==
==13239== For counts of detected and suppressed errors, rerun with: -v
==13239== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```