

Assignment 02 - Sorted List

Invoke the program by running the executable and placing a zero for every test case you wish to skip followed by the output file. Some examples:

Note that TC2 is in the moderate category while TC3 is in the easy category.

To run tests 1 4, and 5:

```
./pa2_test 1 0 0 4 5 0 0 0 0 0 0 tc145.txt
```

To run tests 11

```
./pa2_test 0 0 0 0 0 0 0 0 0 0 11 tc11.txt
```

Easy:

Test case 1: Create an empty List, create an iterator, and then destroy them

Expected Output: None

Possible Errors' Output:

Error: no list returned by SLCreate

Error: no iter returned by SLCreateIterator

Test case 3: Create a new iterator on an empty list and iterate

Expected Output: None

Possible Errors' Output:

Error: \tTest %d: Iter got %i\n

Moderate:

Test case 2: Insert/remove an item from a null list

Expected Output: None

Error Output:

Incorrect return value for SLRemove, %d instead of 0 on failure

Incorrect return value for SLInsert, %d instead of 0 on failure

Test case 4: SLInsert(list, item) into list twice and check return value:

Expected Output: None

Possible Errors' Output:

Incorrect return value for SLInsert, %d instead of 1 on success

Incorrect return value for SLInsert, %d instead of 0 on failure

Test case 5: SLRemove(list, item) twice and check return value:

Expected Output: None

Possible Errors' Output:

Incorrect return value for SLRemove, %d instead of 1 on success

Incorrect return value for SLRemove, %d instead of 0 on failure

Test case 6: Create and iterate through a list with multiple ints:

Expected Output:

Test6: Iter got 5

Test6: Iter got 4

Test6: Iter got 3

Test6: Iter got 2

Test6: Iter got 1

Possible Errors' Output:

Error: Test6: Iter got %d instead of %d

Error: Test6: Iter reached %d items instead of 5

Test case 7: Create and iterate through a list with multiple strs:

Expected Output:

Test7: Iter got Echo

Test7: Iter got Delta

Test7: Iter got Charlie

Test7: Iter got Bravo

Test7: Iter got Alpha

Possible Errors' Output:

Error: Test7: Iter got %s instead of %s

Error: Test7: Iter reached %d items instead of 5

Test case 8: Create and iterate through a list with multiple doubles:

Expected Output:

Test8: Iter got 4.400000

Test8: Iter got 3.300000

Test8: Iter got 2.200000

Test8: Iter got -1.100000

Test8: Iter got -5.500000

Possible Errors' Output:

Error: Test8: Iter got %f instead of %f

Error: Test8 Iter reached %d items instead of 5

Hard:

Test case 9: Create a lot of iterators on a single list:

Expected Output:

Test9a: Iter got Alpha

Test9b: Iter got Bravo

Test9b: Iter got Alpha

Test9c: Iter got Charlie
Test9c: Iter got Bravo
Test9c: Iter got Alpha

Test9d: Iter got Delta
Test9d: Iter got Charlie
Test9d: Iter got Bravo
Test9d: Iter got Alpha

Test9e: Iter got Charlie
Test9e: Iter got Bravo
Test9e: Iter got Alpha

Error Output:

Error: Test9_: Iter got %s instead of %s
Error: Test9_: Iter reached %d items instead of [1/2/3/4/3]
(respectively)

Test case 10: Check if destructor function is called at the correct time:

Expected Output:

Destroy function should not be called here:
Destroy function should be called here: DESTROY FUNCTION CALLED
Destroy function should not be called here:
Destroy function should be called here: DESTROY FUNCTION CALLED
Destroy function should be called here: DESTROY FUNCTION CALLED

Error Output:

Destroy function should not be called here: DESTROY FUNCTION CALLED
Destroy function should be called here:

Test case 11: Check if iterator correctly handles removed elements:

Expected Output:

Test11a: Iter got 5
Test11a: Iter got 2
Test11a: Iter got 1
Test11b: Iter got 4
Test11b: Iter got 2
Test11b: Iter got 1

Error Output:

Error: Test11_: Iter got %d instead of %d
Error: Test11_: Iter reached %d items instead of 3

Test Case 12 (Leaks):

valgrind --leak-check=full --show-reachable=yes --track-origins=yes
./pa2_test 1 2 3 4 5 6 7 8 9 10 0 tc12.txt

Test Case 13 (Leaks):

```
    valgrind --leak-check=full --show-reachable=yes --track-origins=yes  
./pa2_test 0 0 0 0 0 0 0 0 0 0 11 tc13.txt
```