
Name: Yu Qin

Results Screenshots

=====start of the write-up=====

Q1: string

Test command:

```

1. yuq8@andromeda-30:~/253P/hw/hw2/1_str (ssh)
X ..P_APPS/hw/hw2 (zsh) %1 X ../hw/hw2/1_str (zsh) %2 X yuq8@andromeda-30:~/253P/hw/hw2/1_str (ssh) %3
$ ls
1.log a.out* .DS_Store main.cpp Makefile string* string.dSYM/ strprbrk.cpp test_strtok.cpp
yuq8@andromeda-30 13:09:52 ~/253P/hw/hw2/1_str
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o string
-----Congratulation to you! Successfully compile.
-----./string to run
yuq8@andromeda-30 13:09:55 ~/253P/hw/hw2/1_str
$ valgrind ./string
==3585== Memcheck, a memory error detector
==3585== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3585== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==3585== Command: ./string
==3585==
This is
www.tutorialspoint.com
website
This
a
sample
string
Congratulations! All success!
==3585==
==3585== HEAP SUMMARY:
==3585==   in use at exit: 72,704 bytes in 1 blocks
==3585==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==3585==
==3585== LEAK SUMMARY:
==3585==   definitely lost: 0 bytes in 0 blocks
==3585==   indirectly lost: 0 bytes in 0 blocks
==3585==   possibly lost: 0 bytes in 0 blocks
==3585==   still reachable: 72,704 bytes in 1 blocks
==3585==   suppressed: 0 bytes in 0 blocks
==3585== Rerun with --leak-check=full to see details of leaked memory
==3585==
==3585== For counts of detected and suppressed errors, rerun with: -v
==3585== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 13:10:10 ~/253P/hw/hw2/1_str
$

```

Conclusion: right result, 0 error in memory leak.

Q2: parse

Test command:

```
1. yuq8@andromeda-30:~/253P/hw/hw2/2_parsec (ssh) — 125X40
x ..P_APPS/hw/hw2 (zsh) #1 x ..P_APPS/hw/hw2 (zsh) #2 x yuq8@andromeda-30:~/253P/hw/hw2/2_parsec (ssh) #3
yuq8@andromeda-30 14:08:08 ~/253P/hw/hw2/2_parsec
$ ls
1_demo_output_musicLibrary.cpp 2mydemo_output_musicLibrary.cpp .DS_Store input2 Makefile parseC.dSYM/
1_musicLibrary.cpp 2my_musicLibrary.cpp input main.cpp parseC* tokens
yuq8@andromeda-30 14:08:10 ~/253P/hw/hw2/2_parsec
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o parseC
-----Congratulation to you! Successfully compile.
-----input (./parseC < 1_musicLibrary.cpp > tokens) to parse cpp, or all history command on screen.
-----input (diff 1_demo_output_musicLibrary.cpp tokens) to check the diff to screen, if no output, then right.
yuq8@andromeda-30 14:08:12 ~/253P/hw/hw2/2_parsec
$ valgrind ./parseC < 1_musicLibrary.cpp > tokens
==17058== Memcheck, a memory error detector
==17058== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==17058== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==17058== Command: ./parseC
==17058==
==17058==
==17058== HEAP SUMMARY:
==17058==    in use at exit: 72,704 bytes in 1 blocks
==17058==    total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==17058==
==17058== LEAK SUMMARY:
==17058==    definitely lost: 0 bytes in 0 blocks
==17058==    indirectly lost: 0 bytes in 0 blocks
==17058==    possibly lost: 0 bytes in 0 blocks
==17058==    still reachable: 72,704 bytes in 1 blocks
==17058==    suppressed: 0 bytes in 0 blocks
==17058== Rerun with --leak-check=full to see details of leaked memory
==17058==
==17058== For counts of detected and suppressed errors, rerun with: -v
==17058== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 14:08:23 ~/253P/hw/hw2/2_parsec
$ diff 1_demo_output_musicLibrary.cpp tokens
yuq8@andromeda-30 14:08:29 ~/253P/hw/hw2/2_parsec
$ cat tokens
```

detail results:

```
1. yuq8@andromeda-30:~/253P/hw/hw2/2_parsec (ssh)
$ cat tokens
#
include
<
stdio
.
h
>
int
main
(
int
argc
,
char
*
argv
[
]
)
{
char
my_char
=
'A'
;
for
(
int
i
=
0
;
i
<
1024
;
++
i
)
}
```

```
int
main
(
int
argc
,
char
*
argv
[
]
)
{
char
my_char
=
'A'
;
for
(
int
i
=
0
;
i
<
1024
;
++
i
)
printf
(
"\nHello\n" she said.\n"
)
;
}
yua8@andromeda-30 14:08:32 ~/253P/hw/hw2/2_parsec
$ █
```

Conclusion: right result, 0 error in memory leak.