

Name: Yu Qin

Results Screenshots

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

Lab 1

```
$ cd 1_url/
yuq8@andromeda-30 15:09:28 ~/253P/lab/lab3/1_url
$ ls
demo_test_output.txt .DS_Store main.cpp Makefile myoutput.txt urlParse* urlParse.dSYM/ urltest.txt
yuq8@andromeda-30 15:09:30 ~/253P/lab/lab3/1_url
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o urlParse
-----Congratulation to you! Successfully compile.
-----Run manually by :
-----./urlParse < urltest.txt
yuq8@andromeda-30 15:09:43 ~/253P/lab/lab3/1_url
```

```
$ valgrind ./urlParse < urltest.txt > myoutput.txt
==18975== Memcheck, a memory error detector
==18975== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18975== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==18975== Command: ./urlParse
==18975==
==18975==
==18975== HEAP SUMMARY:
==18975==    in use at exit: 72,704 bytes in 1 blocks
==18975==   total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==18975==
==18975== LEAK SUMMARY:
==18975==    definitely lost: 0 bytes in 0 blocks
==18975==    indirectly lost: 0 bytes in 0 blocks
==18975==    possibly lost: 0 bytes in 0 blocks
==18975==    still reachable: 72,704 bytes in 1 blocks
==18975==         suppressed: 0 bytes in 0 blocks
==18975== Rerun with --leak-check=full to see details of leaked memory
==18975==
==18975== For counts of detected and suppressed errors, rerun with: -v
==18975== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 15:11:34 ~/253P/lab/lab3/1_url
$
```

```

$ diff myoutput.txt demo_test_output.txt
yuq8@andromeda-30 15:12:08 ~/253P/lab/lab3/1_url
$ cat myoutput.txt
domain: www.learnthenet.com
path: web-at-a-glance/url-examples/
<blank line>
scheme: http
domain: my.ebay.com
port: 80
path: ws/eBayISAPI.dll
query_string: MyEbayBeta&MyEbay=&gbh=1&guest=1
fragment_id: myfragment
<blank line>
yuq8@andromeda-30 15:12:11 ~/253P/lab/lab3/1_url
$ █

```

Conclusion: right result, 0 error in memory leak.

Little Notes: Implemented with the function of checking some invalid input. Further valid check could include the valid roman char sequence and times.

Lab 2

```

$ ls
demo_output2.txt demo_output.txt freqCount* freqCount.dSYM/ main.cpp Makefile myoutput.txt wolves.txt
yuq8@andromeda-30 15:07:10 ~/253P/lab/lab3/2_frequency
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o freqCount
^[[A^[[A-----Congratulation to you! Successfully compile.
-----Run manually by :
-----./freqCount wolves.txt
yuq8@andromeda-30 15:07:23 ~/253P/lab/lab3/2_frequency
$ valgrind ./freqCount wolves.txt > myoutput.txt
==18259== Memcheck, a memory error detector
==18259== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18259== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==18259== Command: ./freqCount wolves.txt
==18259==
==18259==
==18259== HEAP SUMMARY:
==18259==    in use at exit: 72,704 bytes in 1 blocks
==18259==   total heap usage: 2 allocs, 1 frees, 73,272 bytes allocated
==18259==
==18259== LEAK SUMMARY:
==18259==    definitely lost: 0 bytes in 0 blocks
==18259==    indirectly lost: 0 bytes in 0 blocks
==18259==    possibly lost: 0 bytes in 0 blocks
==18259==    still reachable: 72,704 bytes in 1 blocks
==18259==           suppressed: 0 bytes in 0 blocks
==18259== Rerun with --leak-check=full to see details of leaked memory
==18259==
==18259== For counts of detected and suppressed errors, rerun with: -v
==18259== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 15:07:35 ~/253P/lab/lab3/2_frequency
$ diff myoutput.txt demo_output.txt
yuq8@andromeda-30 15:07:45 ~/253P/lab/lab3/2_frequency
$ █

```

```

$ diff myoutput.txt demo_output.txt
yuq8@andromeda-30 15:07:45 ~/253P/lab/lab3/2_frequency
$ cat myoutput.txt
' ' 88
'"' 10
',' 28
'-' 1
'.' 7
':' 1
'?' 1
'C' 2
'E' 1
'H' 1
'I' 2
'M' 1
'O' 2
'T' 4
'W' 1
'a' 27
'b' 6
'c' 5
'd' 19
'e' 69
'f' 8
'g' 14
'h' 22
'i' 36
'j' 2
'k' 4
'l' 21
'm' 6
'n' 35
'o' 37
'p' 11
'r' 26
's' 32
't' 35
'u' 11
'v' 6
'w' 6
'y' 13
whitespace characters 105
other characters 18yuq8@andromeda-30 15:07:54 ~/253P/lab/lab3/2_frequency
$ █

```

Conclusion: right result, 0 error in memory leak.

Little Notes:

- 1) Inputs are from a file, thus to avoid the treat of line changing '\n'. Further input form can take this into consideration.

[Lab 3](#)

```
1. yuq8@andromeda-30:~/253P/lab/lab3/3_roman (ssh)
X ..3P_APPS/lab/3 (zsh) %1 X yuq8@andromeda-30:~/253P/lab/lab3/3_roman %2
$ pwd
/home/yuq8/253P/lab/lab3/3_roman
yuq8@andromeda-30 21:17:05 ~/253P/lab/lab3/3_roman
$ ls
input input2 main.cpp Makefile outout outout2 outout_demo roman.dSYM/
yuq8@andromeda-30 21:17:06 ~/253P/lab/lab3/3_roman
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o roman
-----Congratulation to you! Successfully compile.
-----input (./parseCmd < [input] > [outout]) to parse all history command on screen, eg.:
-----./roman < input > outout
yuq8@andromeda-30 21:17:12 ~/253P/lab/lab3/3_roman
$ valgrind ./roman < input2
==6403== Memcheck, a memory error detector
==6403== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==6403== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==6403== Command: ./roman
==6403==
invalid input roman number: IIW
IV => 4
==6403==
==6403== HEAP SUMMARY:
==6403==    in use at exit: 72,704 bytes in 1 blocks
==6403==    total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==6403==
==6403== LEAK SUMMARY:
==6403==    definitely lost: 0 bytes in 0 blocks
==6403==    indirectly lost: 0 bytes in 0 blocks
==6403==    possibly lost: 0 bytes in 0 blocks
==6403==    still reachable: 72,704 bytes in 1 blocks
==6403==    suppressed: 0 bytes in 0 blocks
==6403== Rerun with --leak-check=full to see details of leaked memory
==6403==
==6403== For counts of detected and suppressed errors, rerun with: -v
==6403== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 21:17:18 ~/253P/lab/lab3/3_roman
$
```

```

$ valgrind ./roman < input
==6789== Memcheck, a memory error detector
==6789== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==6789== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==6789== Command: ./roman
==6789==
II => 2
IV => 4
VIII => 8
XVI => 16
XXXII => 32
LXIV => 64
CXXVIII => 128
CCLVI => 256
DXII => 512
MXXIV => 1024
MMXLVIII => 2048
MMMXXCVI => 4096
MMMMMMXCXCII => 8192
==6789==
==6789== HEAP SUMMARY:
==6789==    in use at exit: 72,704 bytes in 1 blocks
==6789==    total heap usage: 1 allocs, 0 frees, 72,704 bytes allocated
==6789==
==6789== LEAK SUMMARY:
==6789==    definitely lost: 0 bytes in 0 blocks
==6789==    indirectly lost: 0 bytes in 0 blocks
==6789==    possibly lost: 0 bytes in 0 blocks
==6789==    still reachable: 72,704 bytes in 1 blocks
==6789==    suppressed: 0 bytes in 0 blocks
==6789== Rerun with --leak-check=full to see details of leaked memory
==6789==
==6789== For counts of detected and suppressed errors, rerun with: -v
==6789== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
yuq8@andromeda-30 21:18:20 ~/253P/Lab/Lab3/3_roman
$

```

Conclusion: right result, 0 error in memory leak.

Little Notes: Implemented with the function of checking some invalid input. Further valid check could include the valid roman char sequence and times.

Hw 3.1

- 1) Create a subdirectory under your hw directory called hw3 and put all files related to this homework in that directory.

```

$ ls
hw0/  hw1/  hw2/
yuq8@andromeda-30 14:51:55 ~/253P/hw
$ mkdir hw3
yuq8@andromeda-30 14:51:58 ~/253P/hw
$ cd hw3
yuq8@andromeda-30 14:52:14 ~/253P/hw/hw3
$ ls
gdb.txt  input  input2  main.cpp  outout  outout2  outout_demo
yuq8@andromeda-30 14:52:15 ~/253P/hw/hw3
$ █

```

```

dhcp-v125-043 :: 253P_APPS/Lab/3 » scp 3_roman/* yuq8@andromeda-30.ics.uci.edu:/home/yuq8/253P/hw/hw3
yuq8@andromeda-30.ics.uci.edu's password:
gdb.txt                                100% 143    12.3KB/s  00:00
input                                  100%  84    24.4KB/s  00:00
input2                                 100%   6     1.2KB/s  00:00
main.cpp                              100% 1291   251.3KB/s  00:00
outout                                 100% 169    61.6KB/s  00:00
outout2                               100%  40    10.2KB/s  00:00
outout_demo                           100% 169    54.5KB/s  00:00
dhcp-v125-043 :: 253P_APPS/Lab/3 » █

```

- 2) Define a Makefile in hw3, the default target should make executable and include gdb debugging information in the executable. Define clean.

```

yuq8@andromeda-30 14:52:15 ~/253P/hw/hw3
$ vi Makefile
yuq8@andromeda-30 14:53:26 ~/253P/hw/hw3
$ cat Makefile
main: main.cpp
    @echo -----compiling main.cpp to create executable program main-----
    g++ -ggdb -std=c++11 main.cpp -o roman
    @echo -----Congratulation to you! Successfully compile.
    @echo -----Run manually by :
    @echo -----"./roman < input > outout"

clean:
    rm -rf roman roman.dSYM roman.o
yuq8@andromeda-30 14:53:33 ~/253P/hw/hw3
$ █

```

- 3) Use the debugger and script.

```
yuq8@andromeda-30 14:58:35 ~/253P/hw/hw3
$ script hw3.script
Script started, file is hw3.script
yuq8@andromeda-30 14:58:45 ~/253P/hw/hw3
$ make clean
rm -rf roman roman.dSYM roman.o
yuq8@andromeda-30 14:58:50 ~/253P/hw/hw3
$ make
-----compiling main.cpp to create executable program main-----
g++ -ggdb -std=c++11 main.cpp -o roman
gdb -----Congratulation to you! Successfully compile.
-----Run manually by :
-----./roman < input > outout
yuq8@andromeda-30 14:58:54 ~/253P/hw/hw3
$ gdb roman
GNU gdb (GDB) 7.12
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from roman...done.
(gdb) b 42
Breakpoint 1 at 0x4008b5: file main.cpp, line 42.
```

```
(gdb) quit
A debugging session is active.

    Inferior 1 [process 14772] will be killed.

Quit anyway? (y or n) y
yuq8@andromeda-30 15:00:37 ~/253P/hw/hw3
$ make clean
rm -rf roman roman.dSYM roman.o
yuq8@andromeda-30 15:00:44 ~/253P/hw/hw3
$ ls
gdb.txt hw3.script input input2 main.cpp Makefile outout outout2 outout_demo
yuq8@andromeda-30 15:00:45 ~/253P/hw/hw3
$ cat Makefile
main: main.cpp
    @echo      -----compiling main.cpp to create executable program main-----
    g++ -ggdb -std=c++11 main.cpp -o roman
    @echo      -----Congratulation to you! Successfully compile.
    @echo      -----Run manually by :
    @echo      -----"./roman < input > outout"

clean:
    rm -rf roman roman.dSYM roman.o
yuq8@andromeda-30 15:00:49 ~/253P/hw/hw3
$ exit
exit
Script done, file is hw3.script
yuq8@andromeda-30 15:00:51 ~/253P/hw/hw3
$ ls
gdb.txt hw3.script input input2 main.cpp Makefile outout outout2 outout_demo
yuq8@andromeda-30 15:00:53 ~/253P/hw/hw3
$
```