HW #1: Data Lab - Manipulating Bits

Yejin Lee and Daeyeon Kim
Architecture & Code optimization (ARC) Lab

September 13th, 2018

Overview

- **In this lab,**
  - You will implement some simple functions
  - ex) byte swapping, negation, type conversion..
  - … by manipulating bits directly!

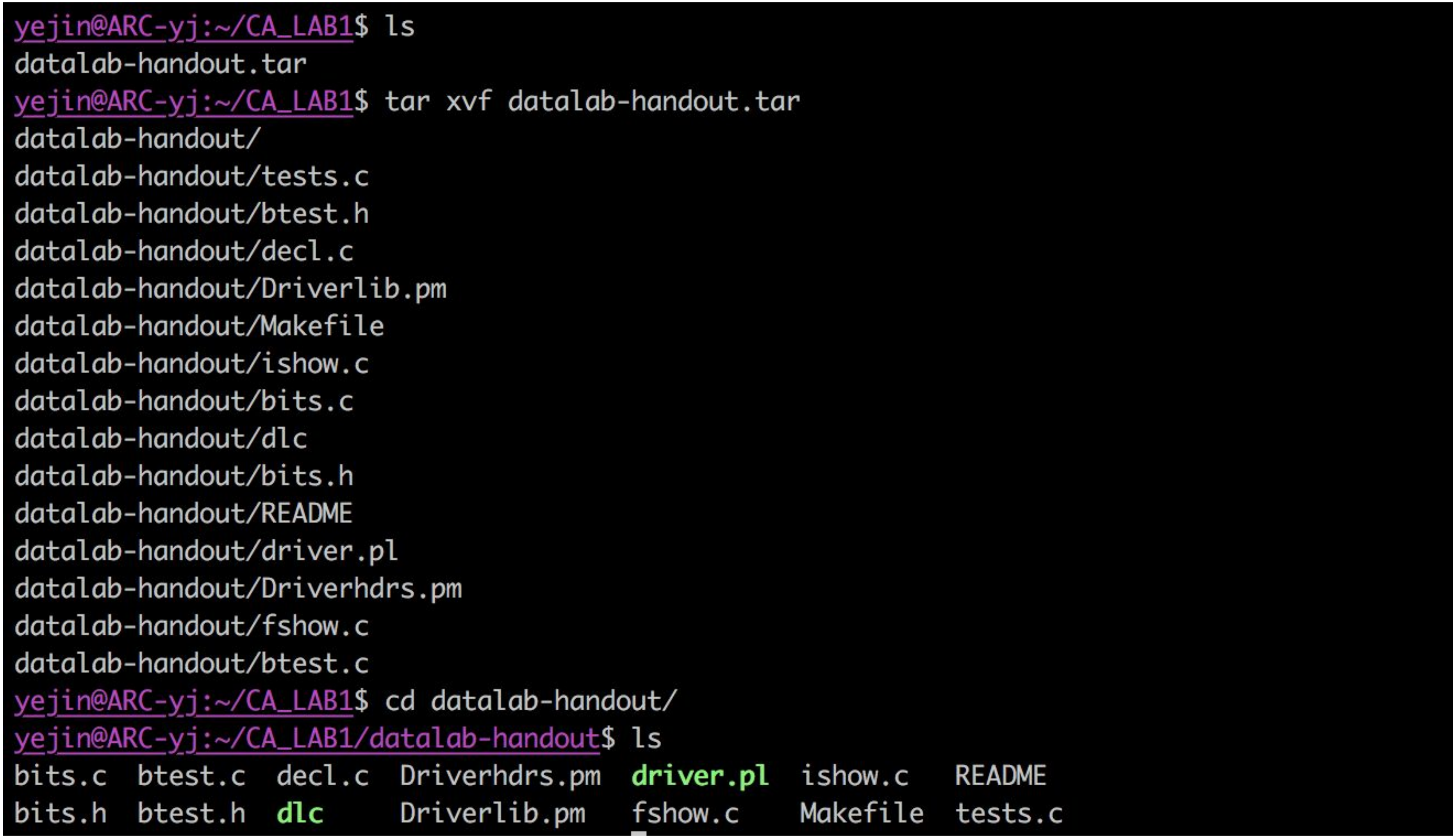
- **By finishing this lab successfully**
  - You will become familiar with **number representations in bit level**

Configuration

- **Linux environment required**
  - ID and PW for Linux machines in software lab (302-311-1) provided
  - Same ID and PW can be used to access Linux server

- **Download and unzip datalab-handout.tar from eTL**

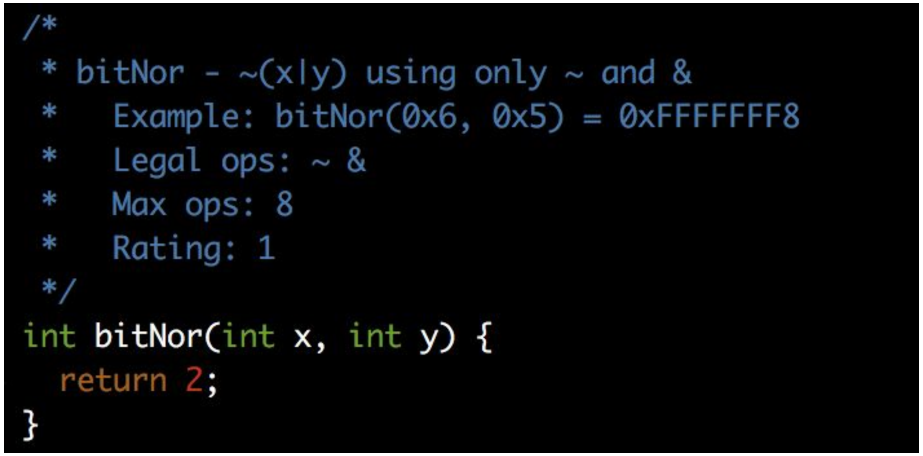
Configuration



```
yejin@ARC-yj:~/CA_LAB1$ ls
datalab-handout.tar
yejin@ARC-yj:~/CA_LAB1$ tar xvf datalab-handout.tar
datalab-handout/
datalab-handout/tests.c
datalab-handout/btest.h
datalab-handout/decl.c
datalab-handout/Driverlib.pm
datalab-handout/Makefile
datalab-handout/ishow.c
datalab-handout/bits.c
datalab-handout/dlc
datalab-handout/bits.h
datalab-handout/README
datalab-handout/driver.pl
datalab-handout/Driverhdrs.pm
datalab-handout/fshow.c
datalab-handout/btest.c
yejin@ARC-yj:~/CA_LAB1$ cd datalab-handout/
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ ls
bits.c  btest.c  decl.c  Driverhdrs.pm  driver.pl  ishow.c   README
bits.h  btest.h  dlc     Driverlib.pm   fshow.c    Makefile  tests.c
```

TODO

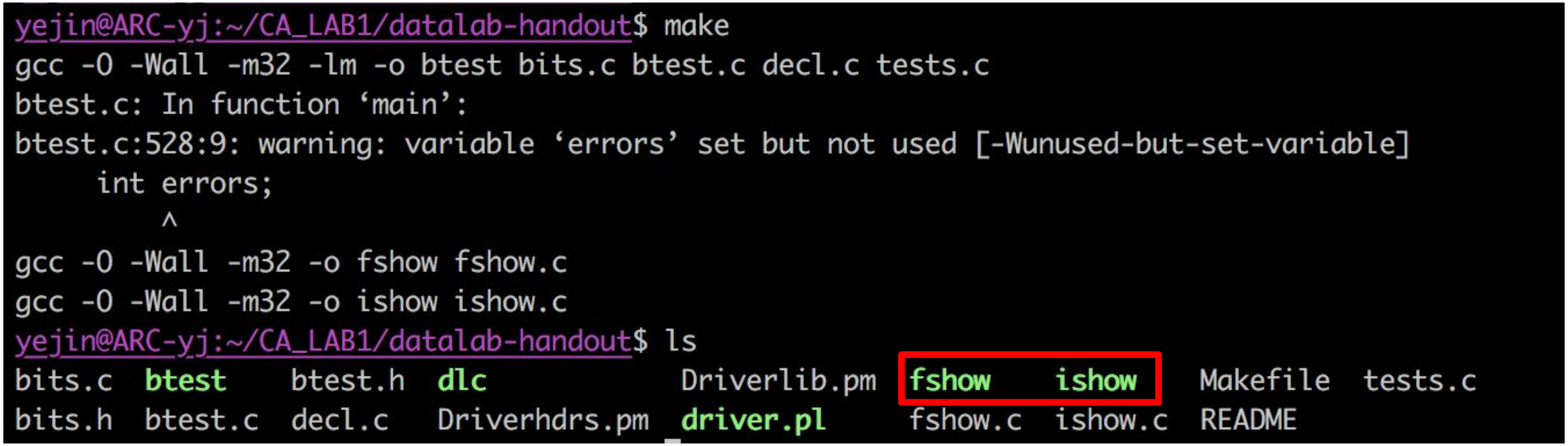
- **Fill in the functions of `bits.c`**



```
/*
 * bitNor - ~(x|y) using only ~ and &
 *   Example: bitNor(0x6, 0x5) = 0xFFFFFFF8
 *   Legal ops: ~ &
 *   Max ops: 8
 *   Rating: 1
 */
int bitNor(int x, int y) {
  return 2;
}
```

- **Be careful: there are rules to follow**
  - Kinds and numbers of operators to use
  - Size of constants
  - Prohibition of using control structs
  - Problems on floating points also have own rules

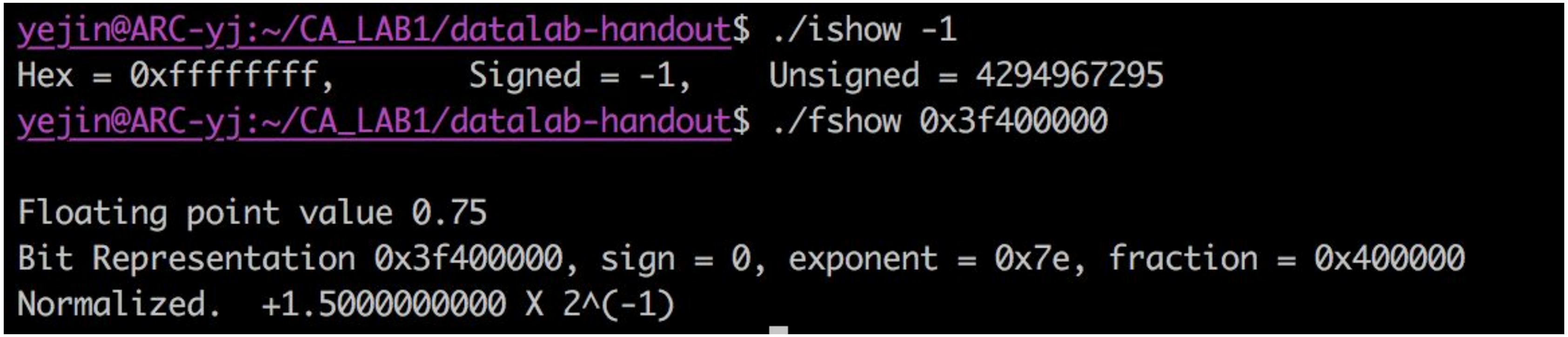
Helpers



```
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ make
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function ‘main’:
btest.c:528:9: warning: variable ‘errors’ set but not used [-Wunused-but-set-variable]
     int errors;
         ^
gcc -O -Wall -m32 -o fshow fshow.c
gcc -O -Wall -m32 -o ishow ishow.c
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ ls
bits.c  btest    btest.h  dlc            Driverlib.pm  fshow    ishow    Makefile  tests.c
bits.h  btest.c  decl.c   Driverhdrs.pm  driver.pl     fshow.c  ishow.c  README
```

Helpers

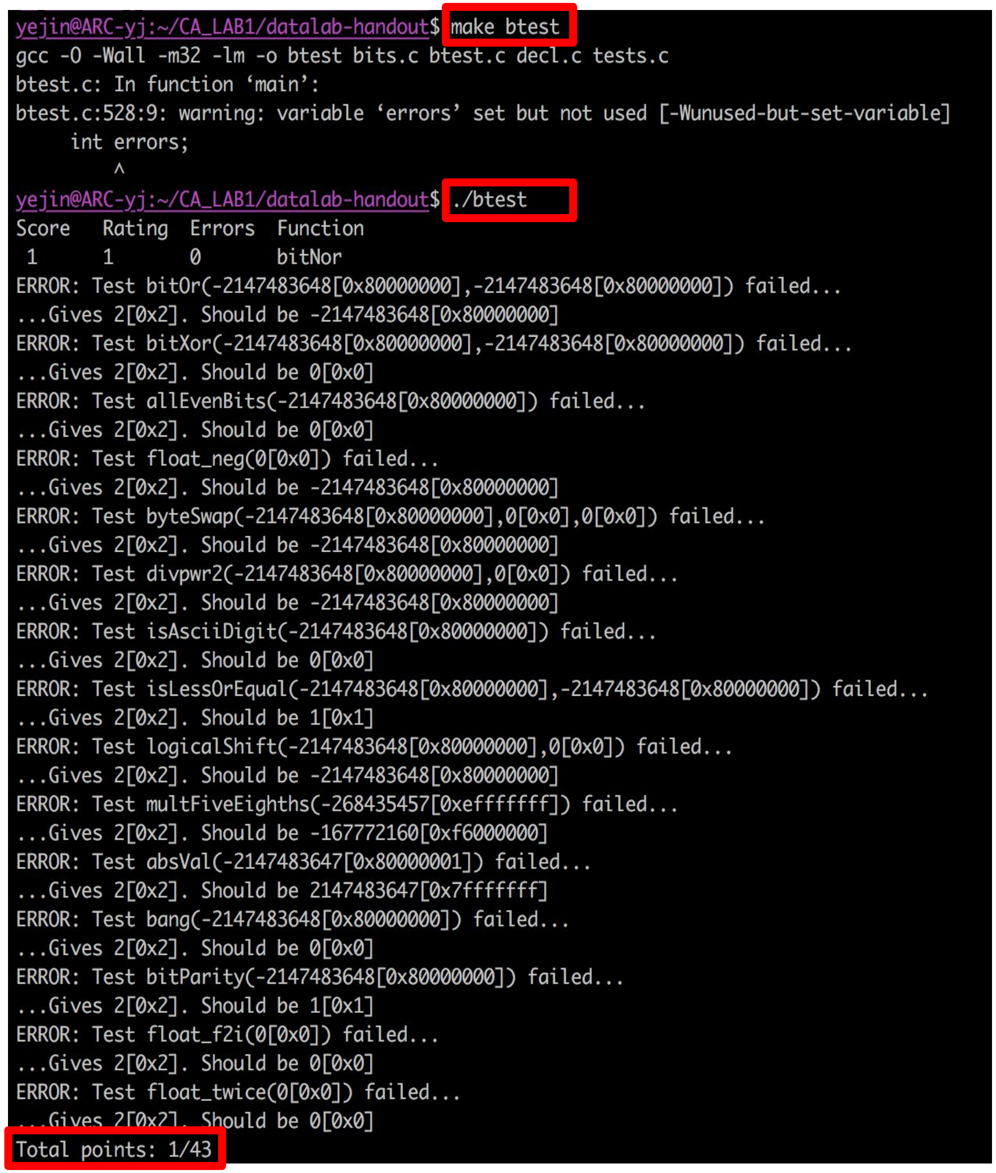
- `./dlc`
  - Check whether you followed all the rules correctly
- `./ishow & ./fshow`
  - Given hex representation, show int/unsigned int/float value
  - Given int/unsigned int/float value, show hex representation



```
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ ./ishow -1
Hex = 0xffffffff,       Signed = -1,    Unsigned = 4294967295
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ ./fshow 0x3f400000

Floating point value 0.75
Bit Representation 0x3f400000, sign = 0, exponent = 0x7e, fraction = 0x400000
Normalized.  +1.5000000000 X 2^(-1)
```

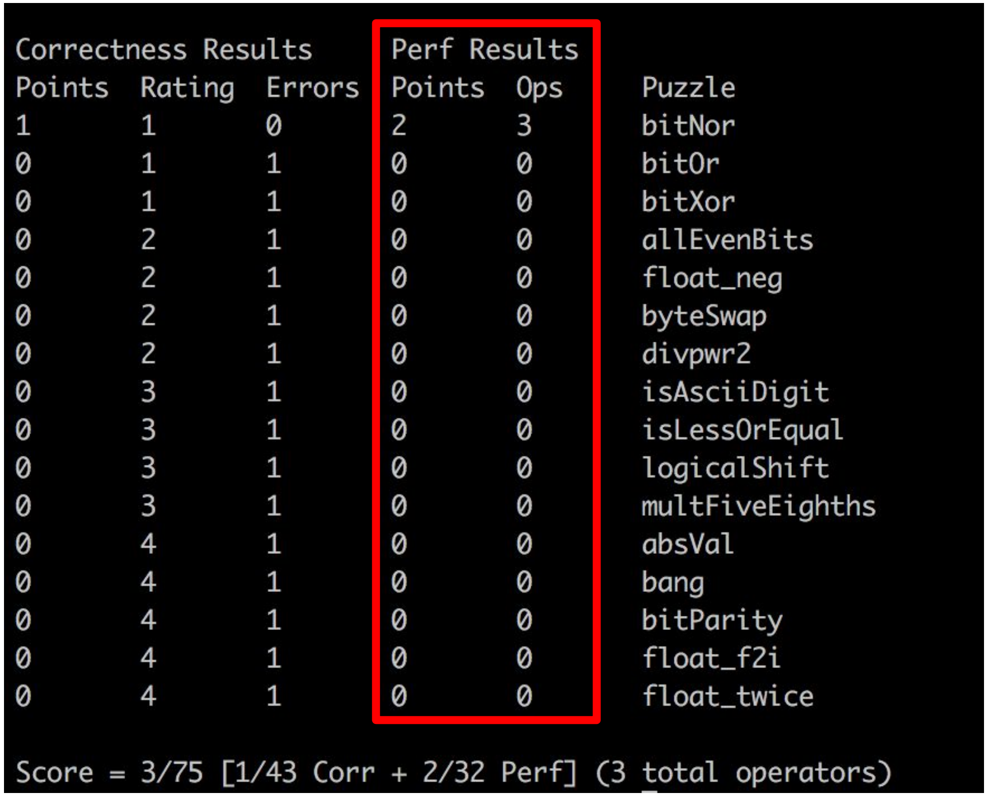
Helpers



```
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ make btest
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
btest.c: In function ‘main’:
btest.c:528:9: warning: variable ‘errors’ set but not used [-Wunused-but-set-variable]
     int errors;
         ^
yejin@ARC-yj:~/CA_LAB1/datalab-handout$ ./btest
Score   Rating  Errors  Function
 1      1       0       bitNor
ERROR: Test bitOr(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test bitXor(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test allEvenBits(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test float_neg(0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test byteSwap(-2147483648[0x80000000],0[0x0],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test divpwr2(-2147483648[0x80000000],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test isAsciiDigit(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test isLessOrEqual(-2147483648[0x80000000],-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 1[0x1]
ERROR: Test logicalShift(-2147483648[0x80000000],0[0x0]) failed...
...Gives 2[0x2]. Should be -2147483648[0x80000000]
ERROR: Test multFiveEighths(-268435457[0xefffffff]) failed...
...Gives 2[0x2]. Should be -167772160[0xf6000000]
ERROR: Test absVal(-2147483647[0x80000001]) failed...
...Gives 2[0x2]. Should be 2147483647[0x7fffffff]
ERROR: Test bang(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test bitParity(-2147483648[0x80000000]) failed...
...Gives 2[0x2]. Should be 1[0x1]
ERROR: Test float_f2i(0[0x0]) failed...
...Gives 2[0x2]. Should be 0[0x0]
ERROR: Test float_twice(0[0x0]) failed...
...Gives 2[0x2]. Should be 0[0x0]
Total points: 1/43
```

- **`./btest`**
  - Test your functions correctness (without performance score)
  - After you modify bits.c,
  - **`make btest`**
  - **`./btest`**

Helpers



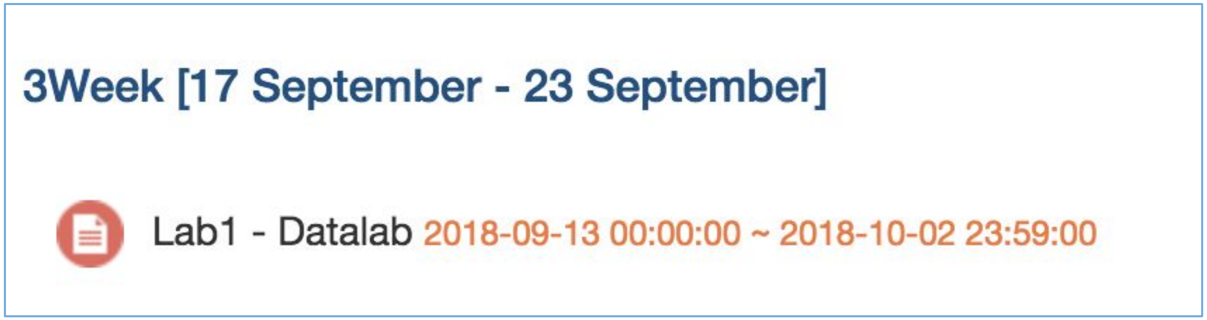
```
Correctness Results     Perf Results
Points  Rating  Errors  Points  Ops     Puzzle
1       1       0       2       3       bitNor
0       1       1       0       0       bitOr
0       1       1       0       0       bitXor
0       2       1       0       0       allEvenBits
0       2       1       0       0       float_neg
0       2       1       0       0       byteSwap
0       2       1       0       0       divpwr2
0       3       1       0       0       isAsciiDigit
0       3       1       0       0       isLessOrEqual
0       3       1       0       0       logicalShift
0       3       1       0       0       multFiveEighths
0       4       1       0       0       absVal
0       4       1       0       0       bang
0       4       1       0       0       bitParity
0       4       1       0       0       float_f2i
0       4       1       0       0       float_twice

Score = 3/75 [1/43 Corr + 2/32 Perf] (3 total operators)
```

- `./driver.pl`
  - Test your functions' correctness with performance score
  - Total Score is 75

Submission

- Submit `bits.c` on eTL



3Week [17 September - 23 September]

Lab1 - Datalab 2018-09-13 00:00:00 ~ 2018-10-02 23:59:00

- Due date: Oct 2 (Tue) 11:59 PM (eTL time stamp)

- Cut-off date: Oct 5 (Fri) 11:59 PM (eTL time stamp)

Grading Policy

- **Different points per problem, from 1 to 4**
  - Each problem has 2 performance points
  - 75 points in total
  - Running `./driver.pl` will let you know your score

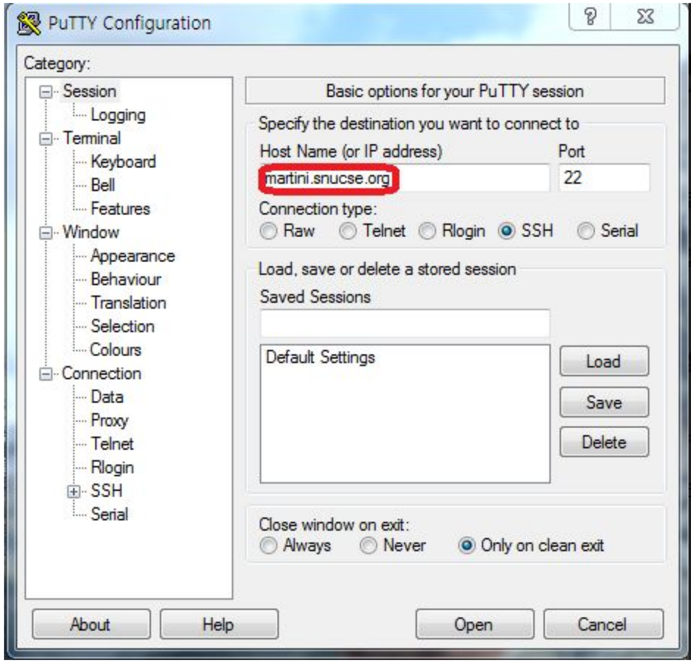
- **Late submission penalty**
  - ~ 24 hrs: -20% of maximum score
  - ~ 48 hrs: -40% of maximum score
  - ~ 72 hrs: -60% of maximum score
  - 72 hrs ~: cut-off (no more submission)
  - Grace Days: no late penalties up to 3 days through this semester

- **Plagiarism**
  - 0 for all assignments (worth of 35% of total grade!)
  - We may use a plagiarism detector program over your codes
  - OK to discuss ideas, but never share your codes in any form

Usage of Server Accounts

- **Log-in computers in software lab**
  - Register student cards and get to 302-311-1
  - States are saved for you

- **Access Linux server (Martini) remotely**



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): martini.snucse.org

Port: 22

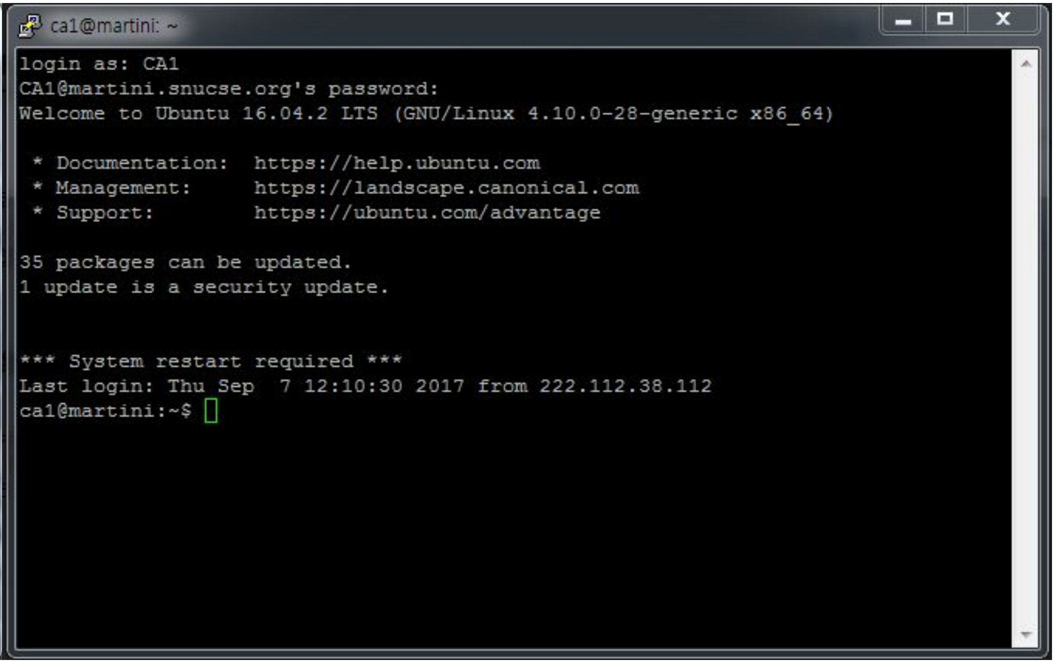
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit



ca1@martini: ~

```
login as: CA1
CA1@martini.snucse.org's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.10.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

35 packages can be updated.
1 update is a security update.


*** System restart required ***
Last login: Thu Sep  7 12:10:30 2017 from 222.112.38.112
ca1@martini:~$ 
```

Q&A