Development Environment & Linux Guide

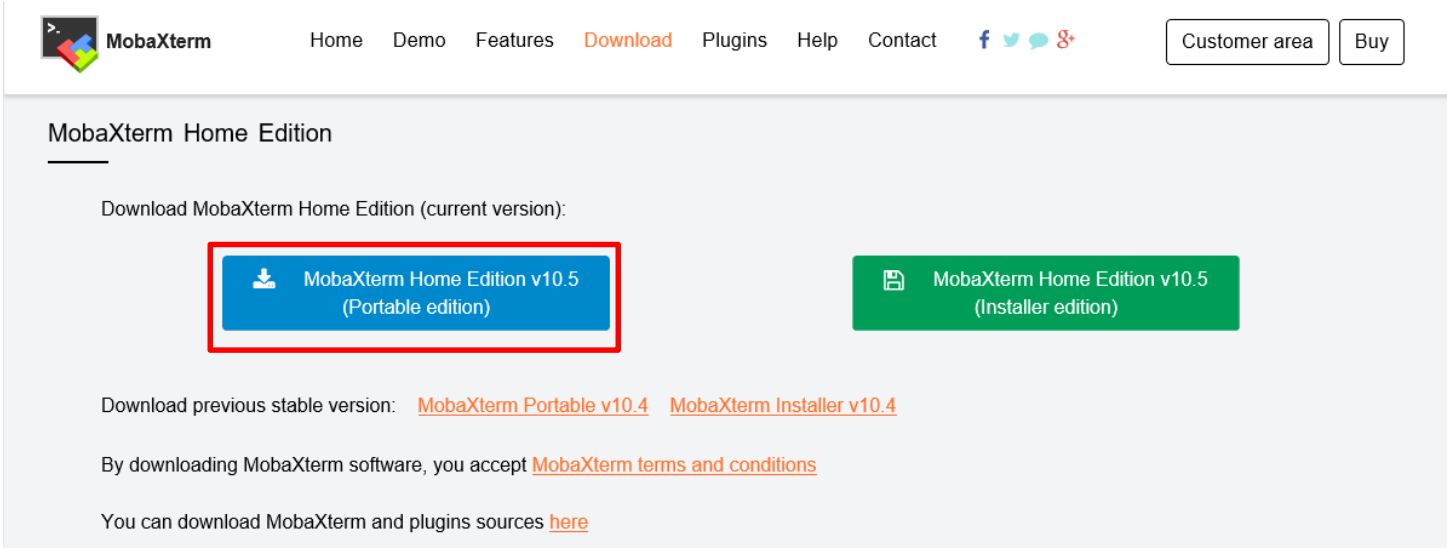
Juwon Lee(jwlee@archi.snu.ac.kr)

School of Computer Science and Engineering

Seoul National University

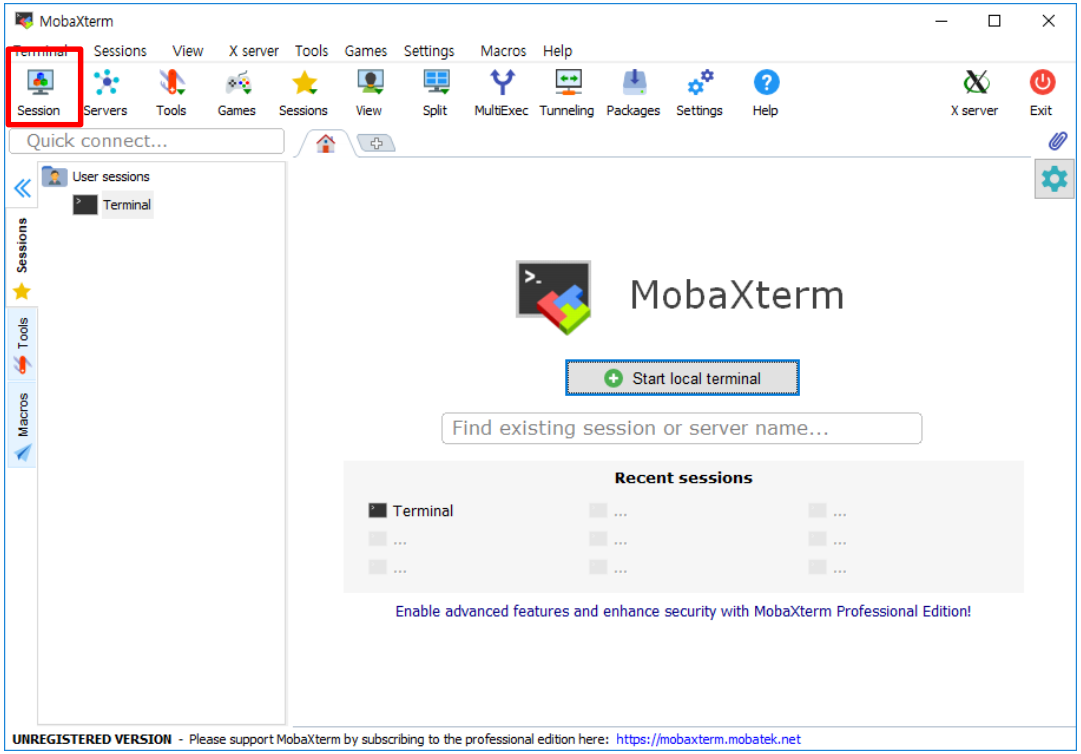
Development Environment

- **MobaXterm**
  - Provide Linux-like environment on Windows.
  - Download and extract MobaXterm at https://mobaxterm.mobatek.net/download-home-edition.html



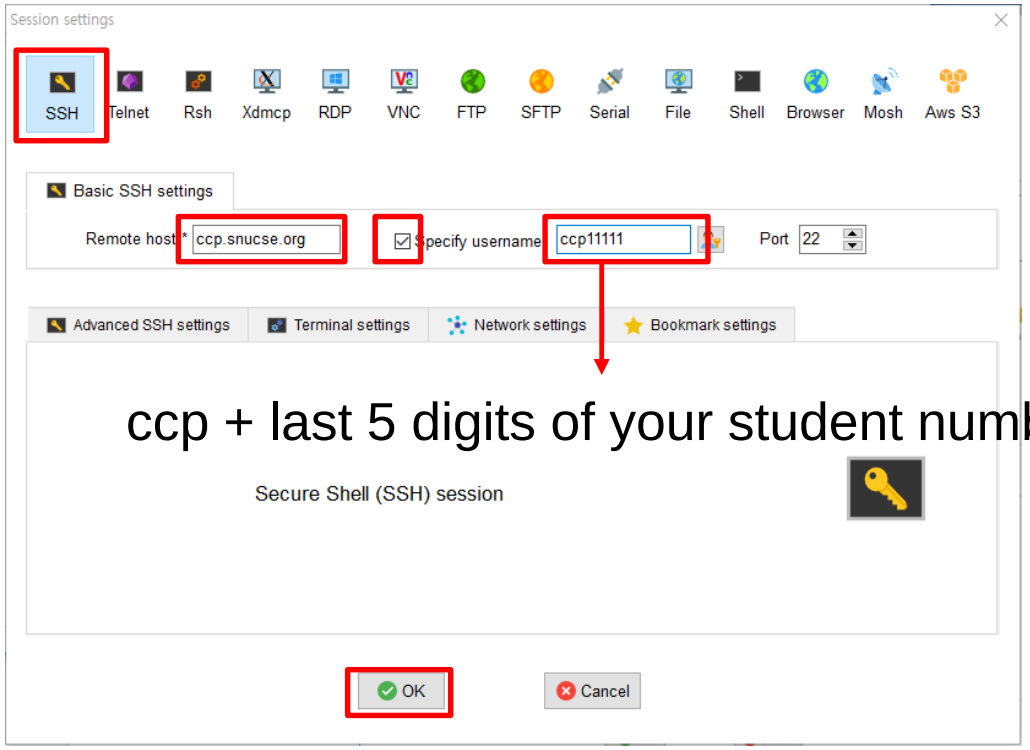
Development Environment

- **SSH**
  - Secure Shell refers to an application or protocol that provides the ability to login, write commands, or copy files to another computer on the network.



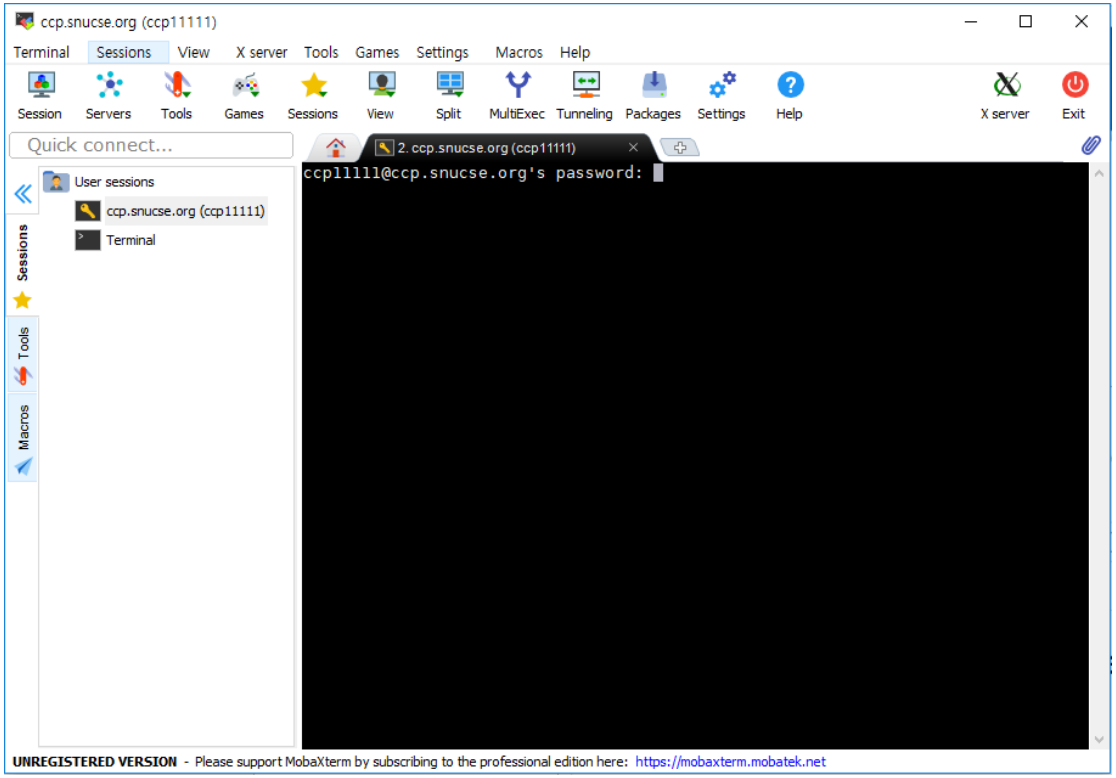
Development Environment

- **SSH**
  - Secure Shell refers to an application or protocol that provides the ability to login, write commands, or copy files to another computer on the network.
  - Mac OS : “ssh ccp11111@ccp.snucse.org” on terminal



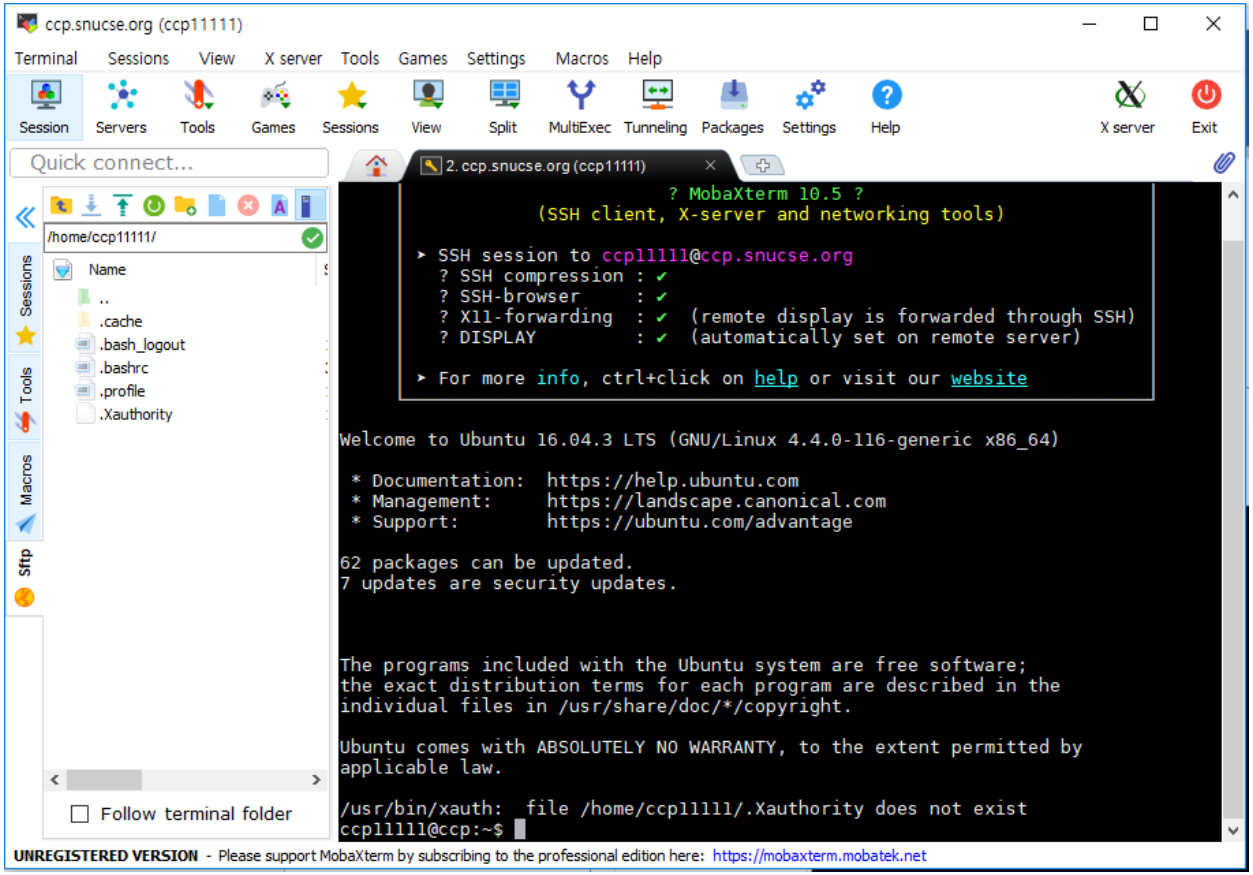
Development Environment

- **SSH**
  - Default password : 1234



Development Environment

- **SSH**
  - It's your personal workspace!



Linux Commands

- **passwd**
  - First thing you have to do after first login.
  - Change password.
  - Type your current password(1234) first and type your new password twice.

```
ccp11111@ccp:~$ passwd
Changing password for ccp11111.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Linux Commands

- **ls**
  - Print the list of files and directories of working directory.

```
ccpllllll@ccp:~$ ls
dir  test
```

- **cd <dest>**
  - Change working directory to destination directory.
  - If the destination directory is “..”, then change to parent directory.
    - ‘.’ means current directory.

```
ccpllllll@ccp:~$ ls
dir  test
ccpllllll@ccp:~$ cd dir
ccpllllll@ccp:~/dir$ cd ..
ccpllllll@ccp:~$
```

Linux Commands

- **pwd**
  - Print working directory.
  - First '/' means root directory.

```
ccpl1111@ccp:~$ pwd
/home/ccpl1111
ccpl1111@ccp:~$ cd dir
ccpl1111@ccp:~/dir$ pwd
/home/ccpl1111/dir
```

- **mkdir <name>**
  - Make new directory.

```
ccpl1111@ccp:~$ ls
dir  test
ccpl1111@ccp:~$ mkdir new_dir
ccpl1111@ccp:~$ ls
dir  new_dir  test
ccpl1111@ccp:~$ cd new_dir/
ccpl1111@ccp:~/new_dir$ pwd
/home/ccpl1111/new_dir
```

Linux Commands

- **rmdir <name>**
  - Remove empty directory.

```
ccp11111@ccp:~$ ls
dir  new_dir  test
ccp11111@ccp:~$ rmdir new_dir
ccp11111@ccp:~$ ls
dir  test
ccp11111@ccp:~$ rmdir dir
rmdir: failed to remove 'dir': Directory not empty
```

- **rm <name>**
  - Remove file.
  - Remove non-empty directory with “-r” option.

```
ccp11111@ccp:~$ ls
dir  test
ccp11111@ccp:~$ rm test
ccp11111@ccp:~$ ls
dir
ccp11111@ccp:~$ rm dir
rm: cannot remove 'dir': Is a directory
ccp11111@ccp:~$ ls
dir
ccp11111@ccp:~$ rm -r dir
ccp11111@ccp:~$ ls
ccp11111@ccp:~$
```

Linux Commands

- **mv <name> <new_name/dir>**
  - Move file/directory to other directory or change name.

```
ccp11111@ccp:~$ ls
dir  test
ccp11111@ccp:~$ mv test new_test
ccp11111@ccp:~$ ls
dir  new_test
ccp11111@ccp:~$ mv new_test dir
ccp11111@ccp:~$ ls
dir
ccp11111@ccp:~$ cd dir
ccp11111@ccp:~/dir$ ls
new_test
```

- **cp <src> <dest>**
  - Copy file

```
ccp11111@ccp:~$ ls
dir  test
ccp11111@ccp:~$ cp test test_backup
ccp11111@ccp:~$ ls
dir  test  test_backup
```

Linux Commands

- **man <command>**
  - Print manual of command.

```
ccp11111@ccp:~$ man ls
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the  FILEs  (the  current directory by default).  Sort
       entries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them; e.g.,  '--block-size=M'  prints
              sizes in units of 1,048,576 bytes; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with  -lt:  sort  by,  and show, ctime (time of last modification of file
              status information); with -l: show ctime and  sort  by  name;  otherwise:
              sort by ctime, newest first
 Manual page ls(1) line 1/227 15% (press h for help or q to quit)
```

Linux Commands

- **vi <name>**
  - Text editor for programming.

Vi

- **Input mode**
  - Edit text like Notepad.
  - To get into input mode from command mode, type ‘a’ / ‘i’ key.

- **Command mode**
  - Initial mode of vi.
  - Additional functions about editing such as find, replace, save, quit, …
  - To get out from input mode, type ‘esc’ key.

Vi

- **From here, everything is working on command mode.**
- **Save / quit**
  - :w – save
  - :w <name> - save as <name>
  - :q! – quit (No save)
  - :wq – save and quit

Vi

- **Copy / paste**
  - yy – copy current line
  - [n]yy – copy [n] lines from current line
  - p - paste

- **Delete line**
  - dd – delete current line
  - [n]dd – delete [n] lines from current line
  - :<from>,<to> d – delete lines from <from> to <to>
    - <from> and <to> is line number
    - '.' means current line, '$' means last line(ex. ":.,$ d" – delete everything from current line)

Vi

- **Find**
  - /<keyword> - find <keyword>
  - n – find next match
  - N – find previous match
- **Replace**
  - :s/<src>/<dest> - replace <src> to <dest> on current line
  - :<from>,<to> s/<src>/<dest> - replace <src> to <dest> from <from> to <to>
  - :%s/<src>/<dest>/g – replace all <src> to <dest> on file
- **Undo**
  - u – undo last command/modified text
  - It maybe very useful.

Compilation

- **gcc**
  - the GNU Compiler Collection.
  - gcc <file>

- **Execution**
  - ./<file_name>

hello.c

```c
#include <stdio.h>

int main(void)
{
        int a = 1;

        printf("Hello world\n");

        return 0;
}
```

Compilation

- **gcc**
  - the GNU Compiler Collection.
  - gcc <file>

- **Execution**
  - ./<binary_file>

```
ccpl1111@ccp:~$ gcc hello.c
ccpl1111@ccp:~$ ls
a.out  dir  hello.c  test
ccpl1111@ccp:~$ ./a.out
Hello world
```

Debugging

- **gdb**
  - the GNU Project Debugger.
  - gcc -g <file>
  - gdb <binary_file>

```
ccp11111@ccp:~$ gcc -g hello.c
ccp11111@ccp:~$ gdb a.out
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
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-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 a.out...done.
(gdb)
```

Debugging

- **Print source code**
  - list – print next 10 lines
  - list <line_number> - print 10 lines around <line_number>
  - list <function_name> - print 10 lines around <function_name>
  - list - – print previous 10 lines

- **Breakpoint**
  - break <line_number> - set breakpoint on <line_number>
  - break <function_name> - set breakpoint on entrance of <function_name>
  - info break – show the list of breakpoint and watchpoint
  - clear <line_number> - delete breakpoint on <line_number>
  - clear <function_name> - delete breakpoint on entrance of <function_name>

Debugging

- **Watchpoint**
  - watch <condition> - suspend processing when <condition> is met
    - ex. watch a > 1 – suspend processing when the value of variable ‘a’ is larger than 1
  - info watch – show the list of watchpoint
  - delete – delete all breakpoints and watchpoints

- **Start and stop execution**
  - run – start program execution from beginning of the program.
  - continue – continue execution on suspend
  - kill – stop program execution
  - quit – exit GDB debugger

Debugging

- **Line execution**
  - step – execute next line of code, step into a function
  - next – execute next line of code, not enter function

- **Examine variables**
  - print <variable> - print value of <variable>

Project Specification
- Minesweeper -

Juwon Lee(jwlee@archi.snu.ac.kr)

School of Computer Science and Engineering

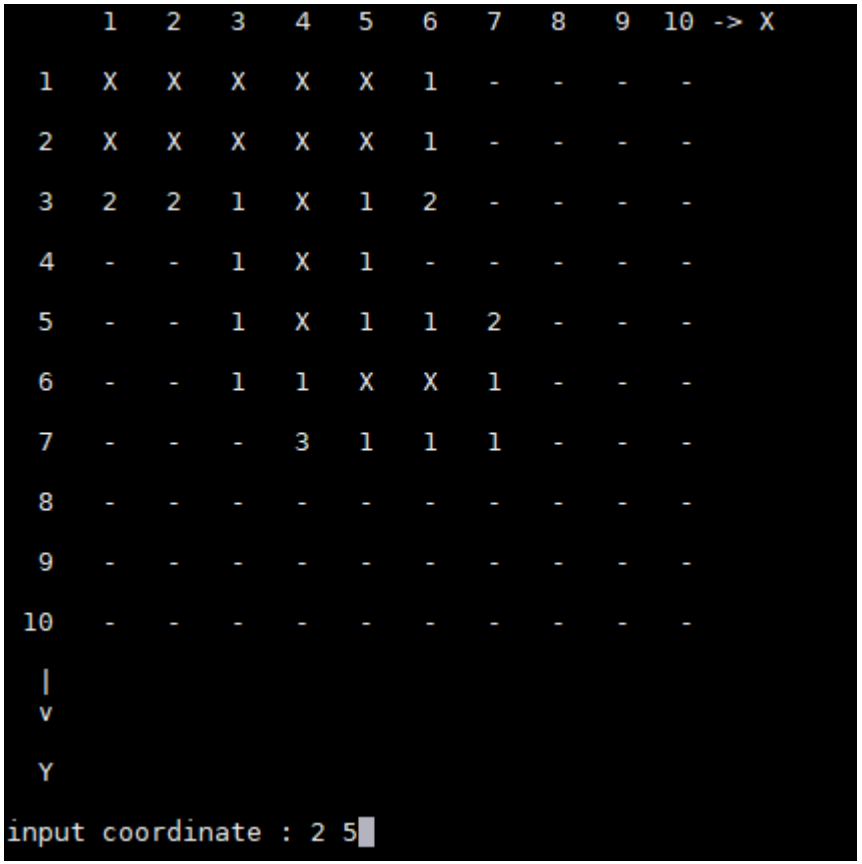
Seoul National University

Minesweeper

- **Rule**
  - Two-dimension array of squares is used and fixed number of mines are planted on squares randomly.
  - The player can sweep squares until the player sweeps all non-mined squares(player win) or the player sweeps a mined square(player lose).
  - If the swept square are not mined and the adjacent squares have one or more mines, the swept square display the number of mines around it.
  - If the swept square are not mined and the adjacent squares have no mine, then the adjacent squares are swept recursively.

Minesweeper

- **Interface**
  - 10*10 array
  - X : no mine around it
  - 1~8 : number of mine around it
  - - : unknown
  - input : coordinate(x, y) to be swept



Minesweeper

- **Variables/Functions**
  - init_board() and show_interface() functions are provided.
  - Your work : fill the main(), sweep() and check_game() functions.

```
int mine_board[BOARD_SIZE][BOARD_SIZE];
/* 0 : non-mined, 1 : mined */

int display_board[BOARD_SIZE][BOARD_SIZE];
/* -1 : no mines around, 0 : unknown, 1~8 : number of mines */

void init_board(); // initialize mine_board by randomly planting fixed number of mines
void show_interface(); // print display_board
int sweep(int x, int y);
/*
 * sweep a square on (x, y). if there is no mines around square on (x, y), recursively sweep adjacent squares
 * return : 1 if player sweeps mined square, else 0
 */

int check_game();
/*
 * check if the player swept all non-mined squares
 * return : 1 if player swept all non-mined squares, else 0
 */
```

- **Please keep this specification and don't modify init_board(), show_interface() function.**

Minesweeper

- **Submission**
  - Compress your source code with name of P_<student_number>.zip
    - ex. P_2018-11111.zip
  - Send your file to jwlee@archi.snu.ac.kr
  - Mail title should be the same as file name except .zip

  - Due date : 2018/06/10 23:59
  - You can submit your code as many times as you want before the due date.

- **If you copy other's program, you will get failing grade.**
  - Don't let other student see your code.

Any Problem or Question

- **E-mail : jwlee@archi.snu.ac.kr**

About Midterm

- **Closed-book, closed-notes.**
- **No cell phones/smart devices/dictionaries allowed.**
- **Prepare your ID card.**