

Practice 07. Arrays and Strings

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National University

vim Configurations

- You can customize vim using a configuration file named “.vimrc”.
- Write the following options in .vimrc and then check vim.

```
set tabstop=4  
set number  
set autoindent
```

- Nice reference
 - <http://dougblack.io/words/a-good-vimrc.html>

More about vim Commands

- There are lot of useful commands in vim.
 - <http://www.fprintf.net/vimCheatSheet.html>
- Counting
 - The commands can be repeated several times by typing the number before the command.
 - **100 dd** will remove 100 lines from the cursor.
- Undo/Redo
 - **u** : undo
 - **CTRL-R** : redo

Copy in Putty

■ Copy

- Drag a area where the texts to be copied are placed.
- Then click the left mouse button.

```
ta@pp: ~  
1 #include <stdio.h>  
2  
3 double inv_11();  
4 double inv_12();  
5 double inv_21();  
6 double inv_22();  
7 int no_inverse();  
8  
9 int main(void)  
10 {  
11  
12     int n; // the number of matrix  
13     int i;  
14     scanf("%d", &n);  
15     for (i= 1; i< n; i++)  
16     {  
17         double a11, a12, a21, a22; // elements of a 2*2 matrix  
18         scanf("%lf%lf%lf%lf", &a11, &a12, &a21, &a22);  
19  
20         if (no_inverse(a11, a12, a21, a22))  
21             printf("Invalid matrix\n");  
22         else  
23         {  
24             // elements of the inverse matrix  
25             double b11, b12, b21, b22;  
26             b11 = inv_11(a11, a12, a21, a22);  
27             b12 = inv_12(a11, a12, a21, a22);  
28             b21 = inv_21(a11, a12, a21, a22);  
29             b22 = inv_22(a11, a12, a21, a22);  
30             printf("%f %f %f %f\n", b11, b12, b21, b22);  
31         }  
32     }  
33     return 0;  
34 }  
35  
36 double inv_11(double a11, double a12, double a21, double a22)
```



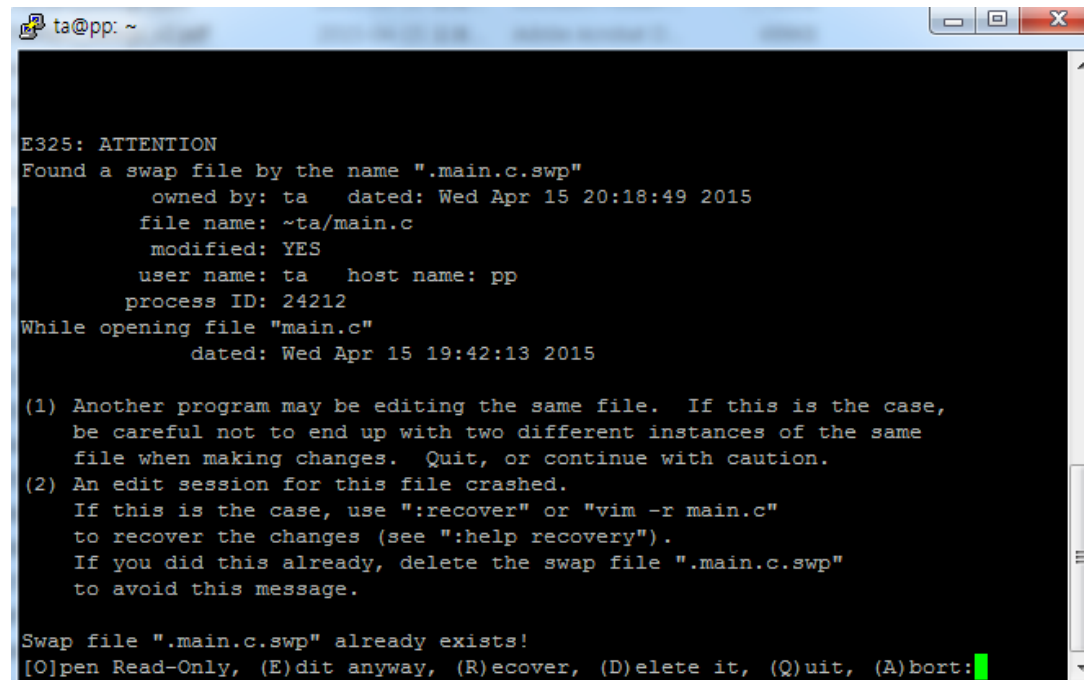
Paste in Putty

- Paste
 - Click the right mouse button from putty.



Swap Files

- The swap files are created when the vim is closed by unexpected accidents.
- Some messages are shown every time when the swap file exists.
- Delete it to avoid annoying messages.

A screenshot of a terminal window titled 'ta@pp: ~'. The terminal displays a Vim error message: 'E325: ATTENTION Found a swap file by the name ".main.c.swp" owned by: ta dated: Wed Apr 15 20:18:49 2015 file name: ~ta/main.c modified: YES user name: ta host name: pp process ID: 24212 While opening file "main.c" dated: Wed Apr 15 19:42:13 2015'. Below this, there are two numbered instructions: (1) Another program may be editing the same file. If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution. (2) An edit session for this file crashed. If this is the case, use ":recover" or "vim -r main.c" to recover the changes (see ":help recovery"). If you did this already, delete the swap file ".main.c.swp" to avoid this message. At the bottom, it says 'Swap file ".main.c.swp" already exists!' followed by a prompt: '[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort:'.

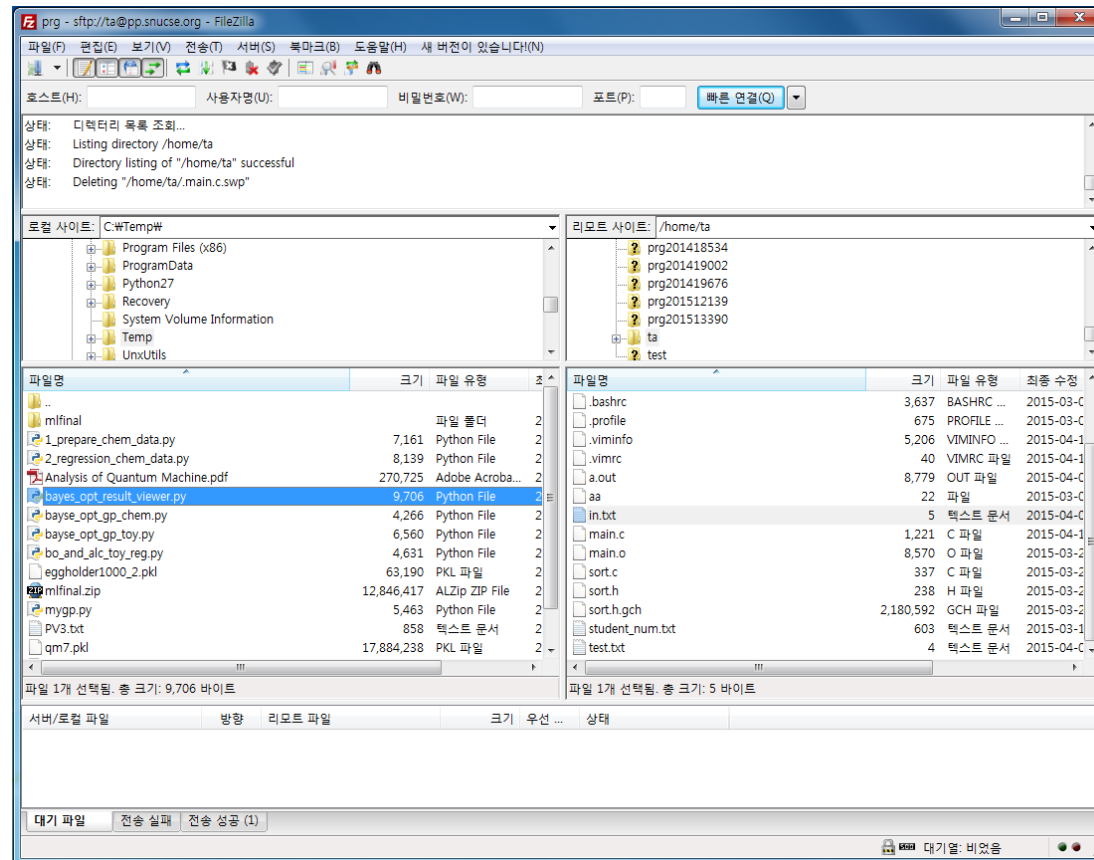
```
ta@pp: ~
E325: ATTENTION
Found a swap file by the name ".main.c.swp"
  owned by: ta   dated: Wed Apr 15 20:18:49 2015
  file name: ~ta/main.c
  modified: YES
  user name: ta  host name: pp
  process ID: 24212
While opening file "main.c"
  dated: Wed Apr 15 19:42:13 2015

(1) Another program may be editing the same file.  If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes.  Quit, or continue with caution.
(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r main.c"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".main.c.swp"
    to avoid this message.

Swap file ".main.c.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort:
```

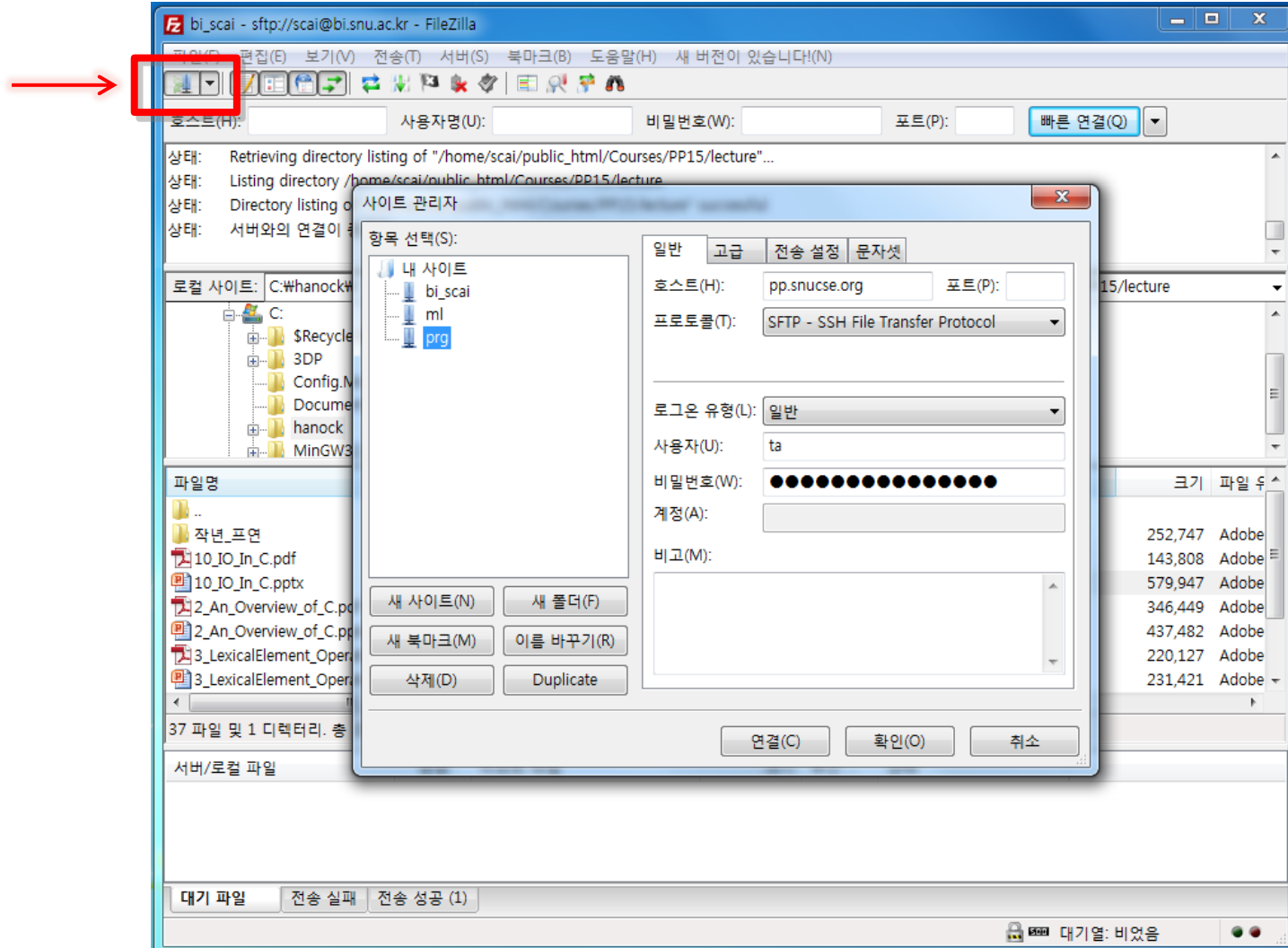
File Transfer via ssh

- There are numerous ways to transfer files via ssh protocol.
- Most popular way to achieve it is to use a program **FileZilla**.



FileZilla

Click here to set the server information.



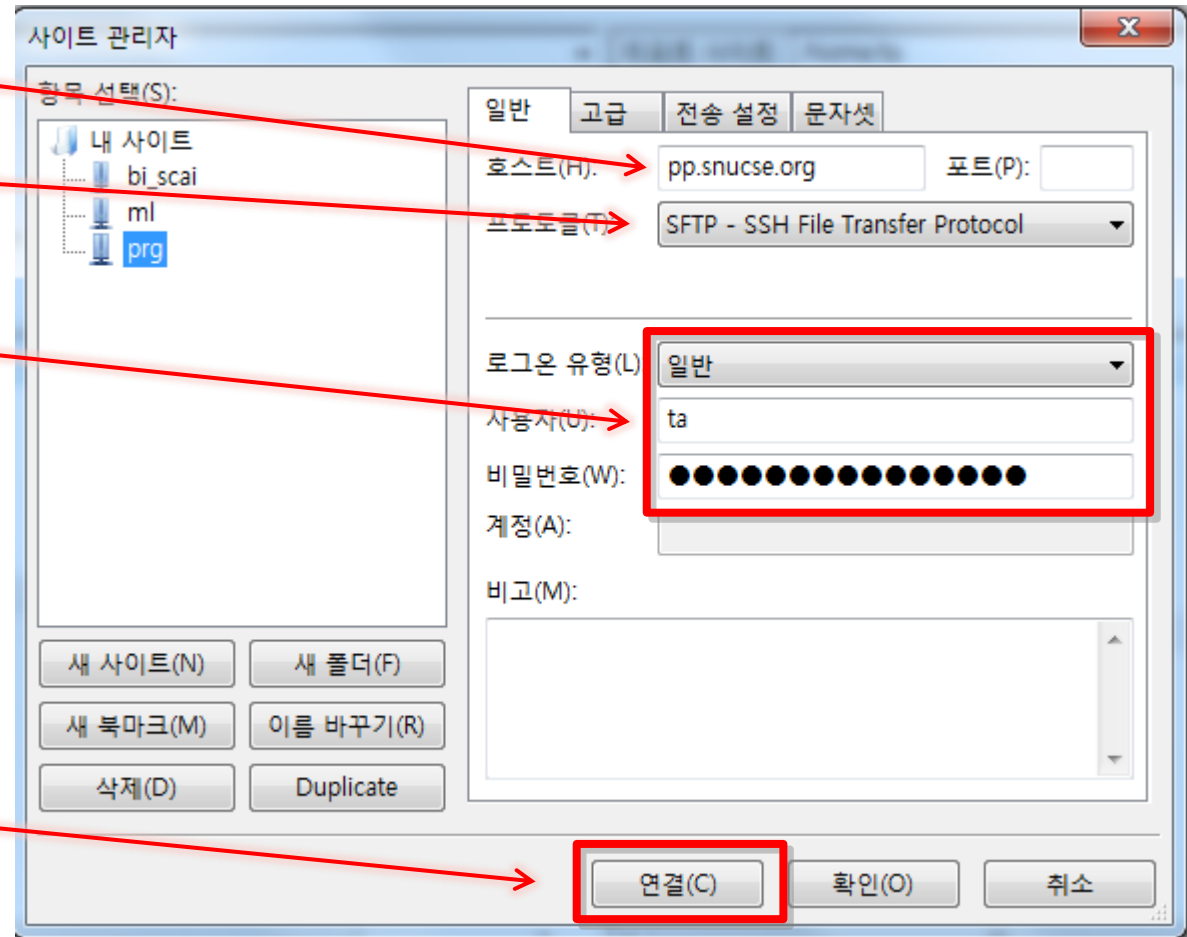
FileZilla

Host : pp.snucse.org

Protocol : SFTP

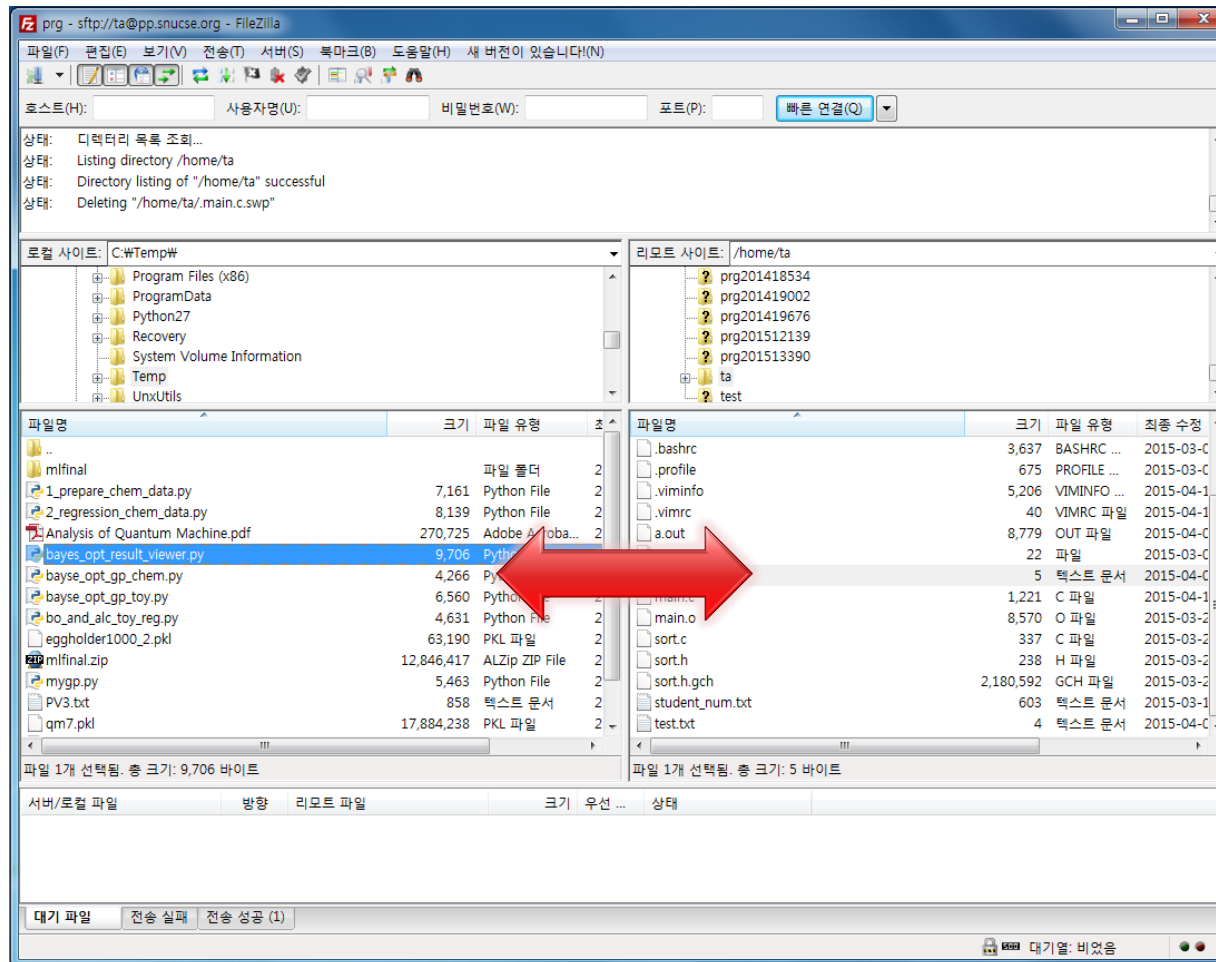
Account : Your user name and password

Connect !!



FileZilla

- Now you can transfer files between the local computer and the server.



Project Guideline 01

- First of the things to be done is to read the board data and construct internal representation in the program.
- For example, you can construct the board data as a character typed multi-dimensional array.
 - **char board[3][4]**
- Secondly, the program should read the pieces in the hands, and then save them in some arrays. For example,
 - **char enemy_hand[8]**
 - **char my_hand[8]**
- Printing the saved data helps for integrity.

Project Guideline 01

- Example

```
int main()
{
    char board[3][4];
    char my_hand[8];
    char enemy_hand[8];

    read_game(board, my_hand, enemy_hand);
    print_game(board, my_hand, enemy_hand);
}
```

Project Guideline 01

- Consider the case when there's none of the pieces in the hands. In such case, fifth and sixth line of the input is empty.
- Use an input file, instead of manually typing the input.

```
root@pp:~# ./a.out < input.txt
```

■ or

```
root@pp:~# cat input.txt | ./a.out
```

- `scanf("%s", str)` will be more easy than `scanf("%c", &ch)`, since `%c` also reads annoying whitespaces.
- `scanf` returns `-1` when there's no more matching data in the input.

Project Guideline 01

- The constants enhance the stability and readability of your program.

```
#define HORI_N 3
#define VERT_N 4
#define MAX_PIECE 8

int main()
{
    char board[HORI_N][VERT_N];
    char my_hand[MAX_PIECE];
    char enemy_hand[MAX_PIECE];
    //...
}
```

Project Guideline 01

- You may need a finishing element in the `my_hand` and `enemy_hand`.
- **`my_hand`**

E	C	C	G	'\0'	'\0'	'\0'	'\0'
---	---	---	---	------	------	------	------

Assignment Submission

- Create a directory named **assignment** in your home directory.
- Create a directory named **07** in your **assignment** directory.
- Put your C files named **p[# of problem].c** for each problem.
 - p01.c
 - p02.c
 - ...
- Due to : **4/22(Wed) 23:59 pm**

Assignment 01 – Bingo

- The input gives 5*5 integers which are one or zero.
- Any consecutive five ones of rows, columns, or diagonal lines are winning patterns of the bingo.
- Output the number of winning patterns.

[Input]

```
1 0 1 0 1
0 1 1 0 1
1 1 1 1 1
0 0 1 1 1
1 1 1 1 1
```

[Output]

5

Assignment 02 – Prefix

- The first line of the input gives a integer N ($N < 20$) and a prefix.
- The next line gives N words.
- Output the number of words that start with the prefix.

[Input]

8 an

ant antibiotic apple analysis acquire banana atmosphere ancestor

[Output]

4