

Practice 08. File IO

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National University

Project Guideline 02

- The function *gets* or *fgets* provide most proper way to read the inputs in our project.
- <http://www.cplusplus.com/reference/cstdio/gets/>
- **char * gets (char * str);**
 - Reads characters from the *standard input* ([stdin](#)) and stores them as a C string into *str* until a **newline character** or the *end-of-file* is reached.
 - On success, the function returns *str*.
If the *end-of-file* is encountered while attempting to read a character, the function returns NULL.

Project Guideline 02

```
#include <stdio.h>

int main()
{
    char buf[256];
    printf ("Insert your full address: ");
    gets(buf);
    printf ("Your address is: %s\n", buf);
    return 0;
}
```

Project Guideline 02

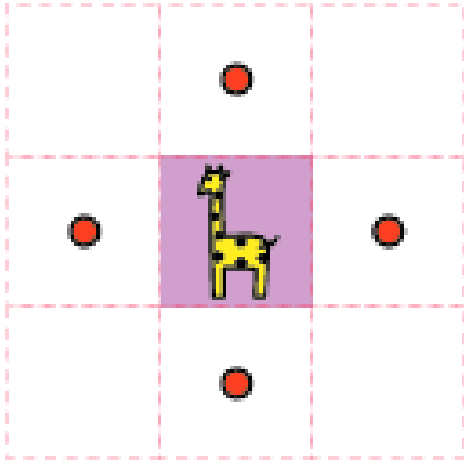
- The inputs always contain six lines where the fifth and sixth line could be empty.

```
1  elg
2  oco
3  oCo
4  GLE
5
6  □
```

- The function *gets* may lead to buffer overflow and it was removed from the recent C standard. It can be replaced by using *fgets* easily.
 - fgets(buf, n, stdin);

Project Guideline 02

- It's tedious to consider every possible movements by hard coding.
- Save the movements in some arrays globally and make a loop over it.



```
// tedious ☹️  
board[x + 1][y] ...  
board[x - 1][y] ...  
board[x][y + 1] ...  
board[x][y - 1] ...
```

```
...  
gx[0] = 0; gy[0] = 1;  
gx[1] = 0; gy[1] = -1;  
gx[2] = 1; gy[2] = 0;  
gx[3] = -1; gy[3] = 0;  
...  
for (i = 0; i < 4; i++)  
{  
    char c = board[x + gx[i]][y + gy[i]];  
}
```

Project Guideline 02

- Making outlined invalid spaces on the board array is more convenient for programming, as it prevents 'out of index error'.

```
char board[3 + 2][4 + 2];  
...  
for (y = 0; y < 6; y++)  
    for (x = 0; x < 5; x++)  
    {  
        if (x == 0 || x == 4 || y == 0 || y == 5)  
            board[x][y] = '\0'; // invalid spaces  
    }
```

\0	\0	\0	\0	\0
\0				\0
\0				\0
\0				\0
\0				\0
\0	\0	\0	\0	\0

Assignment Submission

- Create a directory named **assignment** in your home directory.
- Create a directory named **08** in your **assignment** directory.
- Put your C files named **p[# of problem].c** for each problem.
 - p01.c
 - p02.c
 - ...
- Due to : **4/29(Wed) 23:59 pm**

Assignment 01 – Basic File IO

- Pass a name of a input file as a first argument of your program.
- Output the number of lines in the file only considering ‘\n’.

[Command]

./[your program name] [file name]

ex) ./p01 in.txt

[File Content]

This is a
sample file

[Output]

2

Assignment 02

- This assignment is a part of the project.
- The inputs come from the project.
- Output the state of the game following the format described below.

[Output Format]

Enemy's Hand: [list of the pieces in the enemy's hand]

My Hand: [list of the pieces in my hand]

Board:

[all pieces with the coordinate info.]

[Input]

gol

ooo

oEe

oLo

C

cg

[Output]

Enemy's Hand: cg

My Hand: C

Board:

g A1

l C1

E B3

e C3

L B4

Assignment 03

- This assignment is a part of the project.
- The inputs come from the project.
- Output all possible movements and placements of your giraffes line by line.

[Input]

loe
ooo
oGo
CLo
G
ce

[Output]

G B3 A3
G B3 B2
G B3 C3
G P B1
G P A2
G P B2
G P C2
G P A3
G P C3
G P C4