

# 프로그래밍 연습

실습 #4

## 실습환경

martini.snucse.org

- 자신의 martini 계정/임시계정으로 실습컴퓨터 로그인
- window키 > 왼쪽 'terminal' 켜기
- ssh {id}@martini.snucse.org  
ex)ssh Ppmccl27@martini.snucse.org

# Linux basic

## I. Readability 높은 코드 작성을 위한 vim 설정

```
:set cindent
```

C 문법에 최적화된 자동 indent

```
#include <stdio.h>
```

[설정 전]

```
int main()
```

```
{
```

```
    printf("Hello, World!");
```

```
█
```

```
#include <stdio.h>
```

[설정 후]

```
int main()
```

```
{
```

```
    printf("Hello, World!");
```

```
█
```

# Linux basic

## I. Readability 높은 코드 작성을 위한 vim 설정

```
:set nu
```

Line 마다 줄 번호를 표기하여 에러 발생시 위치 파악 용이

```
#include <stdio.h>
```

[설정 전]

```
int main()
```

```
{
```

```
    printf("Hello, World!");
```

```
    return 0;
```

```
}
```

```
1 #include <stdio.h>
```

```
2
```

```
3 int main()
```

```
4 {
```

```
5     printf("Hello, World!");
```

```
6     return 0;
```

```
7 }
```

[설정 후]

# Linux basic

## I. Readability 높은 코드 작성을 위한 vim 설정

Vim 재시작하여도 설정 유지하기 위해서는 ~/.vimrc 파일 내에 정의 필요

```
ppmccl30@martini:~$ vi ~/.vimrc
```

```
set nu  
set cindent
```

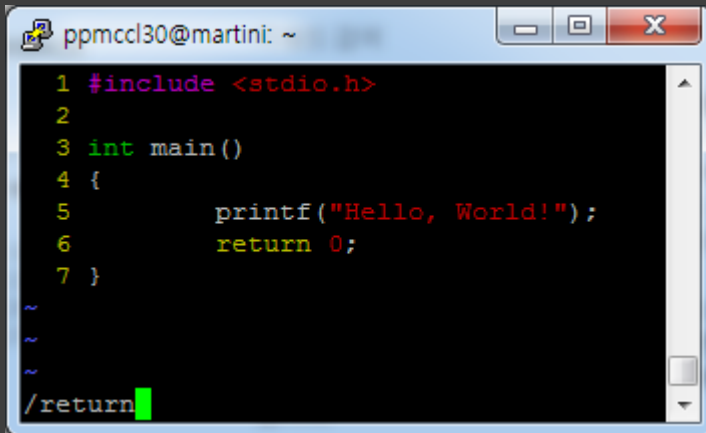
# Linux basic

## II. Vim을 통한 code 내 keyword search 및 line 이동

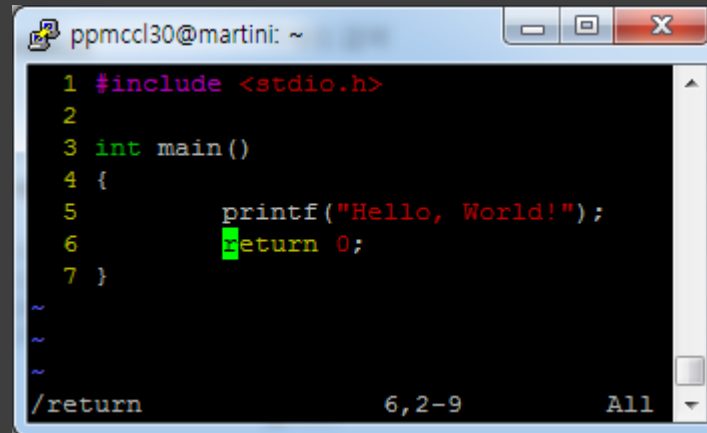
### Keyword search

Code가 길어질 수록 특정 키워드를 찾거나 해당 위치로 이동이 어려워짐

- 기존 Windows 계열 편집기에서는 'Ctrl' + 'F' 를 이용하여 keyword search
- Vim 에서는 normal mode (ESC)에서 '/' 와 'n' 을 이용하여 search 가능



```
ppmcc130@martini: ~  
1 #include <stdio.h>  
2  
3 int main()  
4 {  
5     printf("Hello, World!");  
6     return 0;  
7 }  
~  
~  
~  
/return
```



```
ppmcc130@martini: ~  
1 #include <stdio.h>  
2  
3 int main()  
4 {  
5     printf("Hello, World!");  
6     return 0;  
7 }  
~  
~  
~  
/return 6,2-9 All
```

동일한 keyword를 이용하여 **정방향으로** 계속 search 할 경우 'n'  
동일한 keyword를 이용하여 **역방향으로** 계속 search 할 경우 'Shift' + 'n'

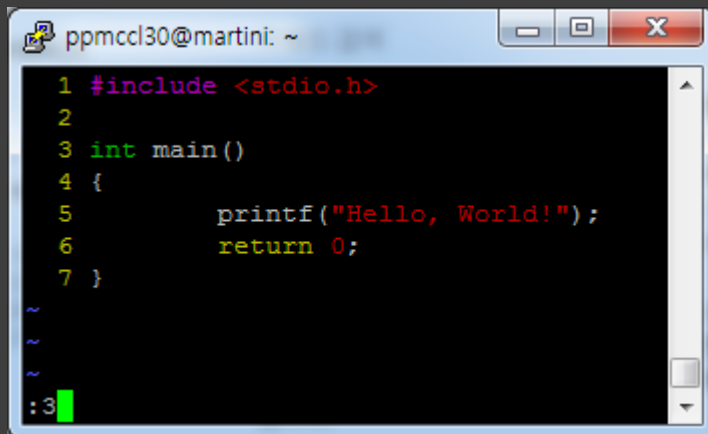
# Linux basic

## II. Vim을 통한 code 내 keyword search 및 line 이동

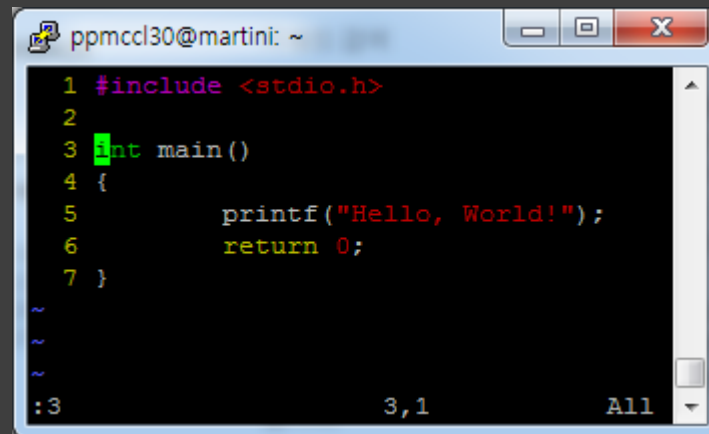
### Line 이동

특정 Line에서 에러가 발생하여 해당 Line으로 이동해야 할 경우 ':' 을 통해 이동하면 바로 이동 가능

- normal mode (ESC)에서 ':' 와 숫자(Line number) 을 이용하여 이동



```
ppmcc130@martini: ~  
1 #include <stdio.h>  
2  
3 int main()  
4 {  
5     printf("Hello, World!");  
6     return 0;  
7 }  
~  
~  
~  
:3
```



```
ppmcc130@martini: ~  
1 #include <stdio.h>  
2  
3 int main()  
4 {  
5     printf("Hello, World!");  
6     return 0;  
7 }  
~  
~  
~  
:3  
3,1 All
```

실습



## 실습1

표준 입력을 받아 해당 입력의 line 수를 계산하는 예제

1. vim ex4-1.c 입력합니다.
2. i를 눌러 입력모드로 변경한 후 다음 코드를 작성해 보세요.

```
#include <stdio.h>
int main(void)
{
    char c;
    int total_line = 0;
    while ((c = getchar()) != EOF)
    {
        if (c == '\n')
            total_line = total_line + 1;
    }
    printf("total line: %d\n", total_line);
    return 0;
}
```

## 실습1

### ➤ 코드분석

```
while ((c = getchar()) != EOF)
```

- 입력으로부터 character들을 순차적으로 입력 받고, EOF(End-of-file) 이 들어오면 while loop를 종료한다.

```
if (c == '\n')
```

```
    total_line = total_line + 1;
```

- 입력된 값이 줄바꿈(\n) 일 경우 total\_line 변수의 값을 1증가시킨다.

```
printf("total line: %d\n", total_line);
```

- 라인수가 저장된 total\_line 변수를 출력한다.

### ✓ 출력 결과

```
ppmcc130@martini:~$ ./ex4-1 < a.txt  
total line: 7
```

## 실습2

문자1개를 입력 받아 해당 문자가 alphabet인지 아닌지를 판단하는 코드 작성 ('q'가 입력되면 alphabet 여부를 판단하지 않고 해당 프로그램 종료)

입력 : 소문자, 대문자, 숫자, 특수문자

```
ppmccl30@martini:~$ ./ex4-2
```

```
>> a
```

```
Alphabet
```

```
>> 1
```

```
No alphabet
```

```
>> N
```

```
Alphabet
```

```
>> -
```

```
No alphabet
```

```
>> q
```

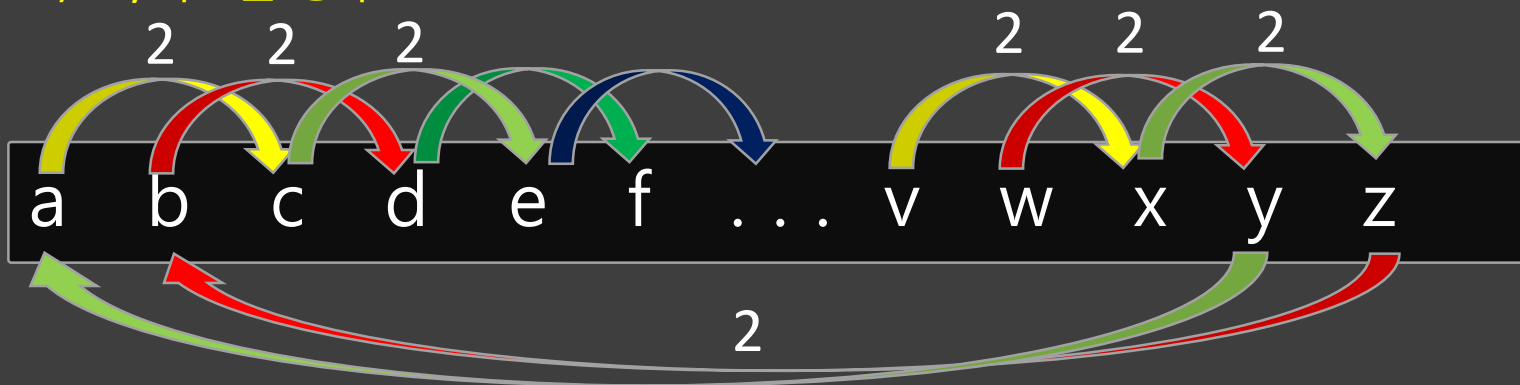
```
ppmccl30@martini:~$
```

### 실습3

암호화 프로그램을 작성한다. 단어(3글자)와 key(0~25)를 입력 받고 암호화 알고리즘을 통해 암호화된 단어를 출력한다. 암호화 알고리즘은 다음과 같다.

※암호화 알고리즘: 해당 알파벳에서 입력 받은 key값 만큼 오른쪽으로 이동

Ex) key가 2인 경우



```
ppmccl30@martini:~$ ./ex4-3
암호화 프로그램입니다.
3글자 단어를 입력하세요.
>> abc
key 값을 입력하세요. (0~25)
>> 2
cde 로 암호화 되었습니다.
```

```
ppmccl30@martini:~$ ./ex4-3
암호화 프로그램입니다.
3글자 단어를 입력하세요.
>> dog
key 값을 입력하세요. (0~25)
>> 25
cnf 로 암호화 되었습니다.
```

# 과제

## 과제1

파일을 입력 재지정(input redirection)을 통해 입력 받아, 파일 안의 다음 글자들의 개수를 세는 프로그램을 작성하라.

전체 글자 수

알파벳 'a'의 개수

숫자 ('0' ~ '9')의 개수

개행문자 ('\n')의 개수

공백문자 (' ' && '\t')의 개수

```
ppmccl30@martini:~/week4$ ./hw4-1 < a.txt
total characters : 69
a : 5
number : 3
line : 5
space : 3
```

a.txt

```
Hello, World!
ABCDEFGHJKLMN
Apple135
Banana
Programming Sample Text
```

## 과제2

2개의 양의 정수를 입력받아 +, -, \*, /, % 의 결과를 출력한다.

- 결과 출력시  
소수점이하 자리가 **있을** 경우 : 소수점이하 첫째자리까지만 출력  
소수점이하 자리가 **없을** 경우 : 정수 부분만 출력

```
ppmccl30@martini:~$ ./hw4-2
양의 정수 2개를 입력하세요 : 3 5
3 + 5 = 8
3 - 5 = -2
3 * 5 = 15
3 / 5 = 0.6
3 % 5 = 3
ppmccl30@martini:~$ ./hw4-2
양의 정수 2개를 입력하세요 : 6 3
6 + 3 = 9
6 - 3 = 3
6 * 3 = 18
6 / 3 = 2
6 % 3 = 0
ppmccl30@martini:~/week4$
```