

# **C Programming (CSE2035)**

## **(Chap10. Strings 2)**



---

**Sungwon Jung, Ph.D.**

**Bigdata Processing & DB LAB**

**Dept. of Computer Science**

**Sogang University**

**Seoul, Korea**

**Tel: +82-2-705-8930**

**Email : [jungsung@sogang.ac.kr](mailto:jungsung@sogang.ac.kr)**



## 실습 1

---

- 10글자 이내의 짧은 문장을 입력 받고 그 문장을 뒤집은 새로운 문장을 만든다. 만약 두 문장이 동일하다면, 이전 문장을 출력하고 그렇지 않다면 이전 문장에 새로운 문장을 덧붙인 문장을 출력하라.
  - 다음 함수를 사용하도록 하시오.
  - *int strlen(const char \*s);*
  - *int strcmp(const char \*s1, const char \*s2);*
  - *char \*strcpy(char \*dest, const char \*src);*
  - *char \*strcat(char \*dest, const char \*src);*



## 실습 1

---

- 실행결과

```
[yjh@mclab ch11]$ ./a.out
Please insert a sentence : qwertrewq
qwertrewq
[yjh@mclab ch11]$
[yjh@mclab ch11]$ ./a.out
Please insert a sentence : 123456789abcdefg
123456789aa987654321
[yjh@mclab ch11]$
```



## 실습 2

- 텍스트 파일인 “data”파일을 읽고, 여기서 정관사 the가 몇번 반복되는지 출력하시오.
  - data 파일은 하나의 문장으로 이루어져 있다. 따라서 fgets() 함수로 이를 읽어 들일 수 있다.
  - strtok 함수를 사용해서 문장을 단어 단위로 잘라내어 strcmp() 함수로 이를 비교할 것.
  - C언어는 대소문자를 구분하므로 “the”, “The”를 구분한다. 따라서 두 경우 모두를 포함하도록 하라.





## 실습 2

- 실행 예제

```
[yjh@mclab ch11]$ cat data
I follow the Moskva down to Gorky Park listening to the wind of change. An August
summer night soldiers passing by listening to the wind of change. The world is
closing in. Did you ever think that we could be so close, like brothers? The fu
ture's in the air I can feel it everywhere blowing with the wind of change. Take
me to the magic of the moment on a glory night where the children of tomorrow d
ream away. In the wind of change.
[yjh@mclab ch11]$
[yjh@mclab ch11]$ ./a.out
total number is 11
[yjh@mclab ch11]$
```



## 실습 3

strcpy()가 문자열의 끝까지 무조건 복사하는 반면, strncpy()는 자신이 원하는 부분에서 원하는 길이만큼 복사가 가능하나, strncpy()에도 여전히 여러 가지 제한이 존재한다.

즉, 함수 `char* strncpy(char *to_string, const char *from_string, int size);`에 대하여,

- 1) `to_string = NULL`이면 segfault가 발생한다.  
(이는 모든 `string.h` 함수에 동일하게 발생한다)
- 2) `size < strlen(from_string)`이면 `to_string` 으로 NULL문자(`\0`)가 복사되지 않는다.
- 3) `size ≥ strlen(to_string)`이면 segfault가 발생한다.
- 4) `size > strlen(from_string)`이면 `from_string`의 범위를 벗어나서 비정상 동작한다.

이번 실습에서는 이러한 제한들을 해결하기 위한 새로운 함수를 작성한다.

```
char* new_strncpy(char **to_string, char *from_string, int size);
```



## 실습 3

`char* new_strncpy(char **to_string, const char *from_string, int size);` 의 인자는 기존 함수 `char* strncpy(char *to_string, const char *from_string, int size);` 의 인자와 동일하다.

`new_strncpy`는 다음과 같은 기능을 가진다.

- 1) `to_string = NULL`이어도 Segmentation Fault없이 정확히 복사된다.
- 2) `to_string`은 `size`에 관계없이 `from_string`의 `[0, size-1]`인 substring을 가진다. 단, `size`가 `from_string`의 길이를 넘어서는 경우 (예 - `from_string = "ABC"`, `size = 5` 등)에는 정확히 `from_string`이 복사되도록 한다.
- 3) 그 외의 사항은 `strncpy()`와 동일하다.
- 4) `size = 0`이면 `strcpy()`와 동일한 기능을 가진다. 즉, 문자열의 끝까지 복사한다.

`from_string`은 `NULL`이 아니라고 가정한다. 또한, 전역 변수의 사용을 금지한다.

# 실습 3

Example)

```
vore@nlpsag:~$ !.
./a.out
input string : I need Openlab!
input-size = 6
before p = (null)
after p = I need
vore@nlpsag:~$
```

다음의 경우에도 에러 없이 정상적으로 동작해야 한다.

```
char *p2 = (char*) malloc(sizeof(char) * 2);
new_strncpy(&p2, "HELLO WORLD!", 5);
char *p3 = (char*) malloc(sizeof(char) * 10);
new_strncpy(&p3, "I need OpenLab!" + 7, 10);

printf("p2 = %s\n", p2);
printf("p3 = %s\n", p3);
```

```
p2 = HELLO
p3 = OpenLab!
```

10글자를 모두 복사  
하지 못하는 경우