

[9장 실습]: 문자열을 다루는 함수

※ 문자열을 다루는 함수를 직접 작성해보면서, 문자열과 포인터의 관계에 대해 공부해본다. 각 함수를 정의하여 프로그램을 완성하시오. (단, 제시된 함수의 원형은 변경할 수 없으며, 문자열을 다루는 어떠한 라이브러리 함수도 사용하지는 않는다.)

※ 모든 문자열 함수는 널을 기준으로 연산이 수행되어야 한다.

int my_strlen(const char *p);

- 문자 배열 또는 문자열 상수(이하 문자열)를 전달받아 문자열의 길이를 반환하는 함수
- 단, 널문자는 세지 않는다.

void my_strcpy(char *p1, const char *p2);

- 두 문자열을 전달받아 두 번째 문자열을 첫 번째 문자열에 복사하는 함수

int my_strcmp(const char *p1, const char *p2);

- 전달받은 두 문자열을 비교하여 같으면 1, 다르면 0을 반환한다.

void makeLower(char p[]); //포인터의 증가연산을 이용할 것

- 문자배열을 전달받아 소문자로 변환해주는 함수(대문자 이외의 문자는 변환되지 않는다)

void makeReverse(char p[]);

- 문자배열을 전달받아 역으로 변환해주는 함수(함수 내부에서 임시 배열을 선언하지 않는다)

void strSwap(char *p1, char *p2); // 포인터의 산술 연산을 이용할 것

- 전달받은 두 문자열의 내용을 서로 교환하는 함수(함수 내부에서 임시 배열을 선언하지 않는다)

void swap(char *a, char *b);

- 두 문자의 내용을 교환하는 함수(makeReverse() 함수와 strSwap() 함수에서 호출된다)

```
int main() {
    char str1[SIZE], str2[SIZE], str3[SIZE];

    printf("문자열(str1) 입력: ");
    gets_s(str1, SIZE);
    printf("\nstr1의 길이 : %d\n", my_strlen(str1));

    my_strcpy(str2, str1);
    makeLower(str2);
    printf("\nstr1을 소문자로 변환 = %s\n", str2);

    my_strcpy(str3, str1);
    makeReverse(str3);
    printf("\nstr1의 역순 = %s\n", str3);

    if (my_strcmp(str1, str2))
        printf("\n%s == %s\n", str1, str2);
    if (!my_strcmp("abc", "abcd"))
        printf("\n\"abc\" != \"abcd\"\n\n");

    my_strcpy(str3, "C is fun");
    printf("----- strSwap() 함수 호출 전-----\n");
    printf("str1 = %s\nstr3 = %s\n", str1, str3);
    strSwap(str1, str3);
    printf("----- strSwap() 함수 호출 후-----\n");
    printf("str1 = %s\nstr3 = %s\n", str1, str3);
    return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
문자열(str1) 입력: C is PowerFull!!

str1의 길이 : 15

str1을 소문자로 변환 = c is powerful!!

str1의 역순 = !!luFrewoP si C

"abc" != "abcd"

----- strSwap() 함수 호출 전-----
str1 = C is PowerFull!!
str3 = C is fun
----- strSwap() 함수 호출 후-----
str1 = C is fun
str3 = C is PowerFull!!
계속하려면 아무 키나 누르십시오 . . .
```