

컴퓨터 프로그래밍 및 실습

10주차. 문자와 문자열

실습 안내

■ 실습 제출 안내

- 솔루션 이름은 "Practice week 10"
- 프로젝트 이름과 소스코드 이름은 Problem1, Problem2, ...
 - 실습1의 프로젝트 이름은 Problem1, 소스코드 이름은 problem1.c
 - 실습 2의 프로젝트 이름은 Problem2, 소스코드 이름은 problem2.c ...
- 솔루션 폴더를 압축하여 **Practice_week10_학번_이름.zip** 으로 제출
- 제출기한: 당일 **19시** 까지

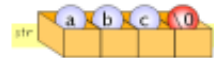
1. 문자열

■ 문자 배열

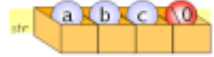
■ 문자열(String)은 문자가 여러 개 모인 것

❖ 문자 배열의 초기화

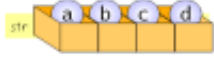
- `char str[4] = { 'a', 'b', 'c', '\0' };`



- `char str[4] = "abc";`



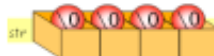
- `char str[4] = "abcdef";`



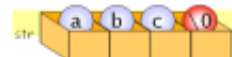
- `char str[6] = "abc";`



- `char str[4] = "";`



- `char str[] = "abc";`



❖ 문자 배열의 출력

```
char str[] = "abc";  
printf("%s", str);
```

```
char str[] = "abc";  
printf(str);
```

1. 문자열

■ 문자열 상수

- String도 배열이기 때문에 포인터 타입이 될 수 있다.
- 하지만 포인터로 문자열 상수를 포인팅 할 시, 수정 불가 - 텍스트 세그먼트 영역

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

int main(void) {
    char str[] = "Hello World!";
    printf("%s\n", str);

    str[0] = 'K';
    printf("%s\n", str);

    char* ptr = "This is char array";
    printf("%s\n", ptr);

    ptr[1] = 'A';    // Error
    printf("%s\n", ptr);

    char* str_ptr = str;
    str_ptr[1] = 'A';    // Valid
    printf("%s\n", str_ptr);

    return 0;
}
```

```
Hello World!
Kello World!
This is char array

D:\git\C-TA\C Programming\P
습니다(코드: -1073741819개)
```

2. 문자 입출력 라이브러리

■ 문자 입출력

입출력 함수	설명
<code>int getchar(void)</code>	하나의 문자를 읽어서 반환한다.
<code>void putchar(int c)</code>	변수 <code>c</code> 에 저장된 문자를 출력한다.
<code>int _getch(void)</code>	하나의 문자를 읽어서 반환한다(버퍼를 사용하지 않음).
<code>void _putch(int c)</code>	변수 <code>c</code> 에 저장된 문자를 출력한다(버퍼를 사용하지 않음).
<code>scanf("%c", &c)</code>	하나의 문자를 읽어서 변수 <code>c</code> 에 저장한다.
<code>printf("%c", c);</code>	변수 <code>c</code> 에 저장된 문자를 출력한다.

2. 문자 입출력 라이브러리

■ 문자 입출력

❖ `_getch()`, `_putch()`

- 버퍼 사용 없음, 에코 없음

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

int main(void)
{
    int ch;
    while( (ch = _getch()) != 'q' )
        _putch(ch);
    return 0;
}
```



❖ 문자 입력 함수 비교

	헤더파일	버퍼사용여부	에코여부	응답성	문자수정여부
<code>getchar()</code>	<code><stdio.h></code>	사용함 (엔터키를 눌러입력됨)	에코	줄단위	가능
<code>_getch()</code>	<code><conio.h></code>	사용하지 않음	에코하지 않음	문자단위	불가능
<code>_getche()</code>	<code><conio.h></code>	사용하지 않음	에코	문자단위	불가능

2. 문자 입출력 라이브러리

■ 실습 1

- x, y가 좌표에서 w a s d 키로 x와 y 방향으로 이동하는 프로그램을 개발해보자.
 - W키가 눌리면 y방향으로 1만큼 이동한다.
 - A키가 눌리면 x방향으로 -1 만큼 이동한다.
 - S키가 눌리면 y방향으로 -1만큼 이동한다.
 - D키가 눌리면 x방향으로 1 만큼 이동한다.
- 각 키가 눌릴 때 마다 좌표가 출력되어야 한다.
- 입력 버퍼를 사용하지 않는다.
 - wasd 가 보이지 않아야 함
- T키가 눌리면 프로그램을 종료해야 한다.

```
0 1
0 2
1 2
2 2
2 1
2 1
2 1
2 0
2 -1
2 -2
1 -2
1 -1
0 -1
-1 -1
0 -1
0 0
-1 0

D:\git\WC-TA\WC Progr
( 0개 )
```

2. 문자 입출력 라이브러리

■ 문자열 입출력 라이브러리

입출력 함수	설명
<code>int scanf("%s", s)</code>	문자열을 읽어서 문자배열 <code>s[]</code> 에 저장
<code>int printf("%s", s)</code>	배열 <code>s[]</code> 에 저장되어 있는 문자열을 출력한다.
<code>char *gets_s(char *s, int size)</code>	한 줄의 문자열을 읽어서 문자 배열 <code>s[]</code> 에 저장한다.
<code>int puts(const char *s)</code>	배열 <code>s[]</code> 에 저장되어 있는 한 줄의 문자열을 출력한다.

2. 문자 입출력 라이브러리

■ 한글과 아스키

- 아스키 코드는 각 문자당 1byte 이기 때문에 char 자료형에 담길 수 있음
- 한글 문자는 주로 UTF-8이나 EUC-KR으로 인코딩 되어 있음
 - UTF-8: 한글 1글자 당 3byte
 - EUC-KR: 한글 1글자 당 2byte
- 따라서 한글은 char 자료형 하나에 담을 수 없다.
- UTF-8의 경우 char형 3개에 한글 하나가 담김
- EUC-KR의 경우 char형 2개에 한글 하나가 담김

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

int main(void) {
    char a1;
    scanf("%c", &a1);

    printf("%c", a1);

    return 0;
}
```

가
?
D:\git

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

int main(void) {
    char arr[10];
    scanf("%s", arr);
    printf("%c%c", arr[0], arr[1]);
    return 0;
}
```

가
가
가

3. 문자 처리 라이브러리

- 문자 검사 라이브러리

함수	설명
isalpha(c)	c가 영문자인가?(a-z, A-Z)
isupper(c)	c가 대문자인가?(A-Z)
islower(c)	c가 소문자인가?(a-z)
isdigit(c)	c가 숫자인가?(0-9)
isalnum(c)	c가 영문자이나 숫자인가?(a-z, A-Z, 0-9)
isxdigit(c)	c가 16진수의 숫자인가?(0-9, A-F, a-f)
isspace(c)	c가 공백문자인가?(' ', '\n', '\t', '\v', '\r')
ispunct(c)	c가 구두점 문자인가?
isprint(c)	c가 출력가능한 문자인가?
iscntrl(c)	c가 제어 문자인가?
isascii(c)	c가 아스키 코드인가?

3. 문자 처리 라이브러리

- 문자 변환 라이브러리

함수	설명
<code>toupper(c)</code>	c를 대문자로 바꾼다.
<code>tolower(c)</code>	c를 소문자로 바꾼다.
<code>toascii(c)</code>	c를 아스키 코드로 바꾼다.

3. 문자 처리 라이브러리

- 실습 2

- 문자열을 입력 받아서 소문자는 대문자로, 대문자는 소문자로 변환하여 출력해보자

3. 문자 처리 라이브러리

- 문자열 처리 라이브러리

함수	설명
strlen(s)	문자열 s의 길이를 구한다.
strcpy(s1, s2)	s2를 s1에 복사한다.
strcat(s1, s2)	s2를 s1의 끝에 붙여넣는다.
strcmp(s1, s2)	s1과 s2를 비교한다.
strncpy(s1, s2, n)	s2의 최대 n개의 문자를 s1에 복사한다.
strncat(s1, s2, n)	s2의 최대 n개의 문자를 s1의 끝에 붙여넣는다.
strncmp(s1, s2, n)	최대 n개의 문자까지 s1과 s2를 비교한다.
strchr(s, c)	문자열 s안에서 문자 c를 찾는다.
strstr(s1, s2)	문자열 s1에서 문자열 s2를 찾는다.

3. 문자 처리 라이브러리

■ 실습 3

- char 배열에 a 부터 z까지 순서대로 담겨있다.
- 사용자가 입력한 알파벳을 기준으로 개행하여 출력한다.
- 출력은 무조건 `printf("%s", ...)`를 사용해야 한다.
- 예시)
 - 입력: d
 - 출력
 - a b c
 - d e f g ... z

3. 문자 처리 라이브러리

■ 문자열 처리 라이브러리

❖ 문자열 토큰(token) 분리

Syntax:

```
예 char s[] = "Hello World";  
char delimit[] = " ";  
char *p = strtok(s, delimit);
```

문자열을 스페이스문자를 사용하여 단어들로 분리한다.

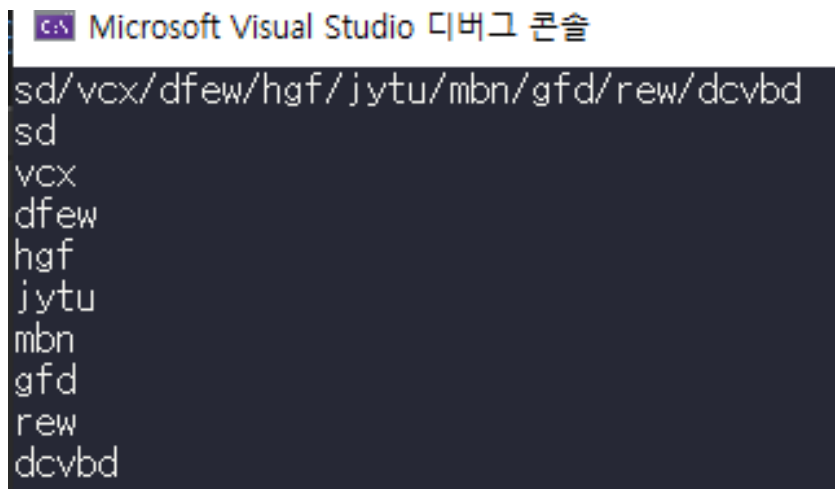
```
char s[] = "Hello World! Hi Hey" ;  
t1 = strtok(s, " ");           // 첫 번째 토큰  
t2 = strtok(NULL, " ");        // 두 번째 토큰  
t3 = strtok(NULL, " ");        // 세 번째 토큰  
t4 = strtok(NULL, " ");        // 네 번째 토큰
```

- 더 이상 분리할 문자열이 없을 경우에는 NULL을 반환한다.

3. 문자 처리 라이브러리

■ 실습 4

- '/'를 구분자로 분리된 문자열을 개행하여 출력해보자



Microsoft Visual Studio 디버그 콘솔

```
sd/vcx/dfew/hgf/jytu/mbn/gfd/rew/dcvbd  
sd  
vcx  
dfew  
hgf  
jytu  
mbn  
gfd  
rew  
dcvbd
```


4. 문자열 수치 변환

❖ 문자열 수치 변환

- 문자열 "36.5"와 수치 값 36.5는 컴퓨터 안에서 상당히 다르게 저장됨
- 문자열을 수치로 변경 또는 수치를 문자열로 변경 가능

함수	설명
sscanf(s, ...)	문자열 s로부터 지정된 형식으로 수치를 읽어서 변수에 저장
sprintf(s, ...)	변수의 값을 형식 지정자에 따라 문자열 형태로 문자배열 s에 저장

4. 문자열 수치 변환

❖ 전용 함수 사용

- 전용함수는 scanf()보다 크기가 작음
- stdlib.h에 원형 정의

함수	설명
<code>int atoi(const char *str);</code>	str을 int형으로 변환한다.
<code>long atol(const char *str);</code>	str을 long형으로 변환한다.
<code>double atof(const char *str);</code>	str을 double형으로 변환한다.

4. 문자열 수치 변환

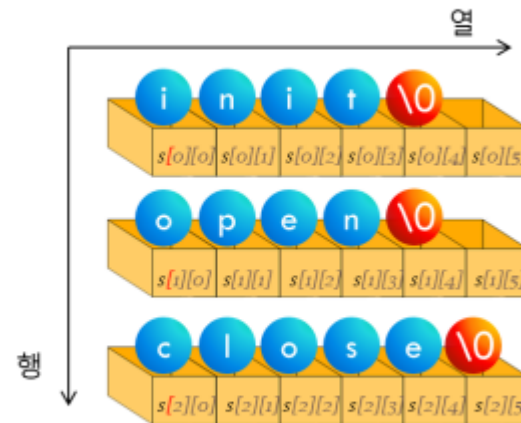
■ 실습 5

- 벡터 덧셈기를 개발해보자.
- $[x1, y1, z1] + [x2, y2, z2]$ 형식으로 입력값이 들어오며, 이를 scanf의 %s 형식으로 읽어온다.
- 연산의 결과값을 출력한다.

5. 문자열 배열

❖ 문자열의 배열

```
char s[3][6] = {  
    "init",  
    "open",  
    "close"  
};
```



5. 문자열 배열

■ 실습 6

- 사용자의 채팅에서 욕설이 포함되어 있다면, *로 변환하여 가리는 클린 채팅 프로그램을 개발해보자.
- 욕설은 문자열 배열에 저장되어 있으며, 아래와 같다.
 - fool
 - prat
 - dog
- 입력 예시
 - fdsaklwic fool cklqwe prat
- 출력 예시
 - fdsaklwic **** cklqwe ****