

10장. 문자열

숭실대학교
전자정보공학부 IT융합전공
담당교수: 권민혜
minhae@ssu.ac.kr

이번 장에서 학습할 내용

- 문자열이 컴퓨터 내부에서 어떻게 표현되는지를 알 수 있다.
- 문자열 입출력 함수들을 사용할 수 있다.
- 문자열을 비교하고 복사할 수 있다.
- 여러 개의 문자열을 저장하고 처리할 수 있다.

인간은 문자를 사용하여 정보를 표현하므로 문자열은 프로그램에서 중요한 위치를 차지하고 있습니다. 이번 장에서는 C에서의 문자열 처리 방법에 대하여 자세히 살펴볼 것입니다.

이번 장에서 만들 프로그램

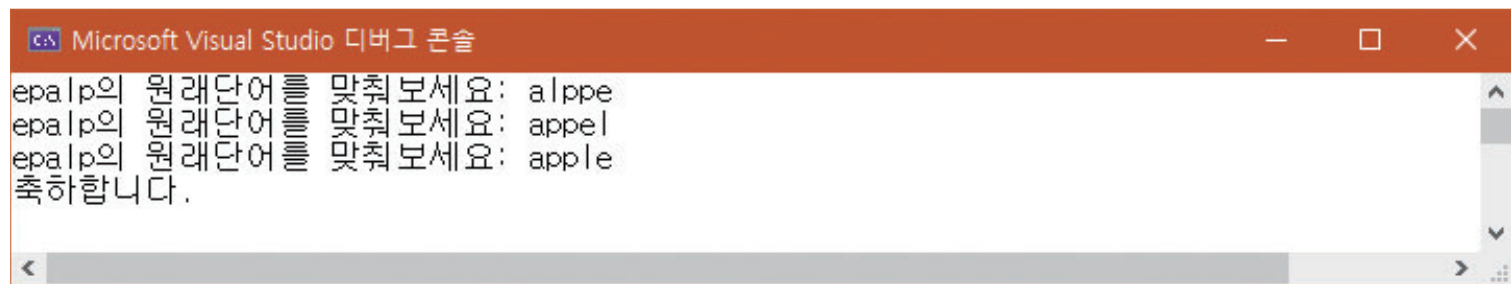
- “행맨” 게임을 작성해보자.



```
C:\# D:\#hello#Debug#hello.exe

문자열을 입력하시오: _____
자리를 추측하시오:
문자열을 입력하시오: ____ a_ _____
자리를 추측하시오:
문자열을 입력하시오: _ee_ a_ _____
자리를 추측하시오: ■
```

- 영어 단어 스크램블 게임을 작성해보자.



```
Microsoft Visual Studio 디버그 콘솔

epalp의 원래단어를 맞춰보세요: alppe
epalp의 원래단어를 맞춰보세요: appel
epalp의 원래단어를 맞춰보세요: apple
축하합니다.
```

- 문자열을 암호화하는 프로그램을 만들어보자.



```
Microsoft Visual Studio 디버그 콘솔

암호화 전의 문자열=This is a house.
암호화 키=123456
암호화 후의 문자열=eZZG^_B$R9]YDAV→羲羲I
```

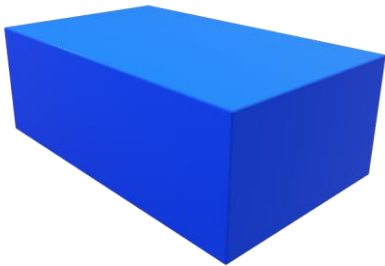
문자의 중요성

- 문자열(string)은 "Hello"와 같이 문자들의 모임이다.

"A"

"Hello"

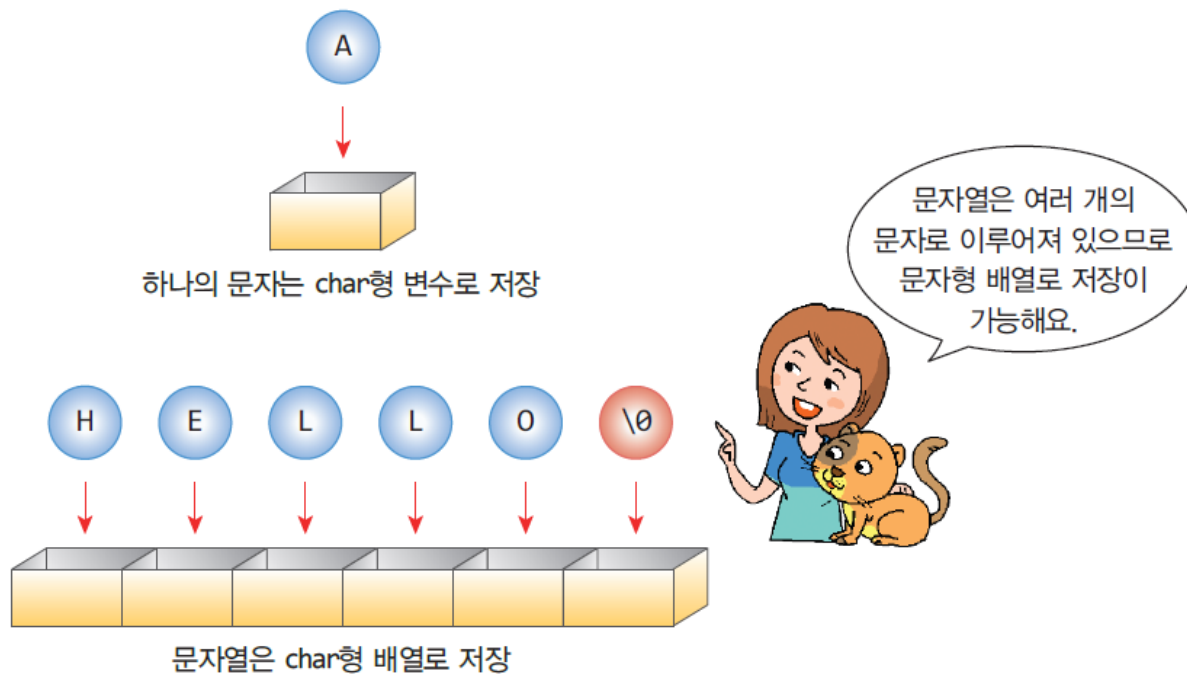
"Mr. Hong"



문자열은 어디에
저장하는 것이
좋을까요?

문자열 표현 방법

```
char str[6] = "Hello";
```



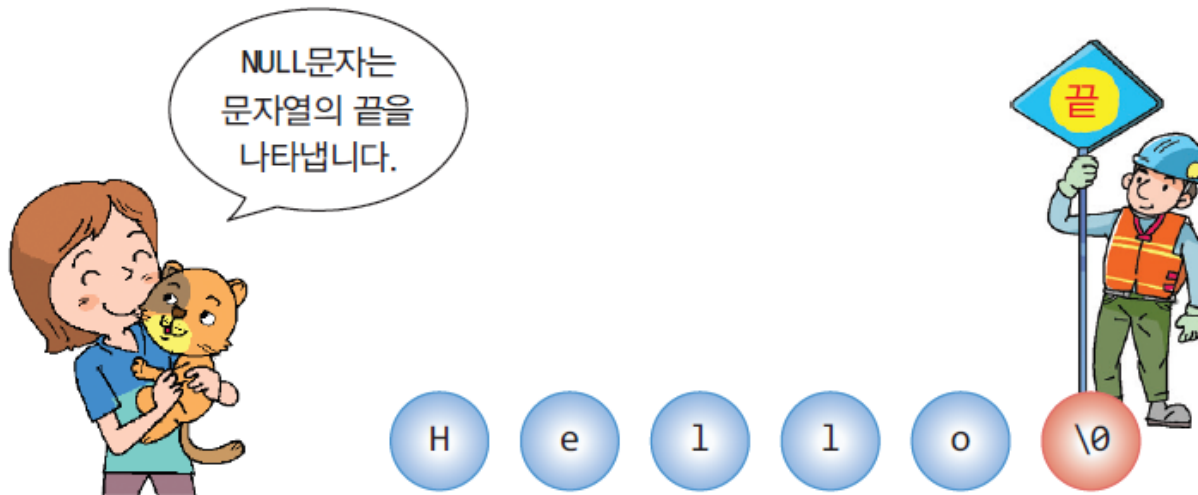
문자표현방법

- 컴퓨터에서는 각각의 문자에 숫자코드를 붙여서 표시한다.
- 아스키코드(ASCII code): 표준적인 8비트 문자코드
 - 0에서 127까지의 숫자를 이용하여 문자표현
- C언어에서는 아스키코드만을 사용한다.

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

NULL 문자

- 문자열의 끝은 반드시 NULL 문자(null character)라는 특수한 값으로 끝나야 한다



```
char str[6] = "Hello";
```

NULL 문자가 필요한 이유

- 사람은 "Hello"만이 의미 있는 문자열이고 "#%?&\$"은 쓰레기 문자라는 것을 알지만 컴퓨터는 어디서부터 어디까지가 의미 있는 문자열인지 알지 못한다.

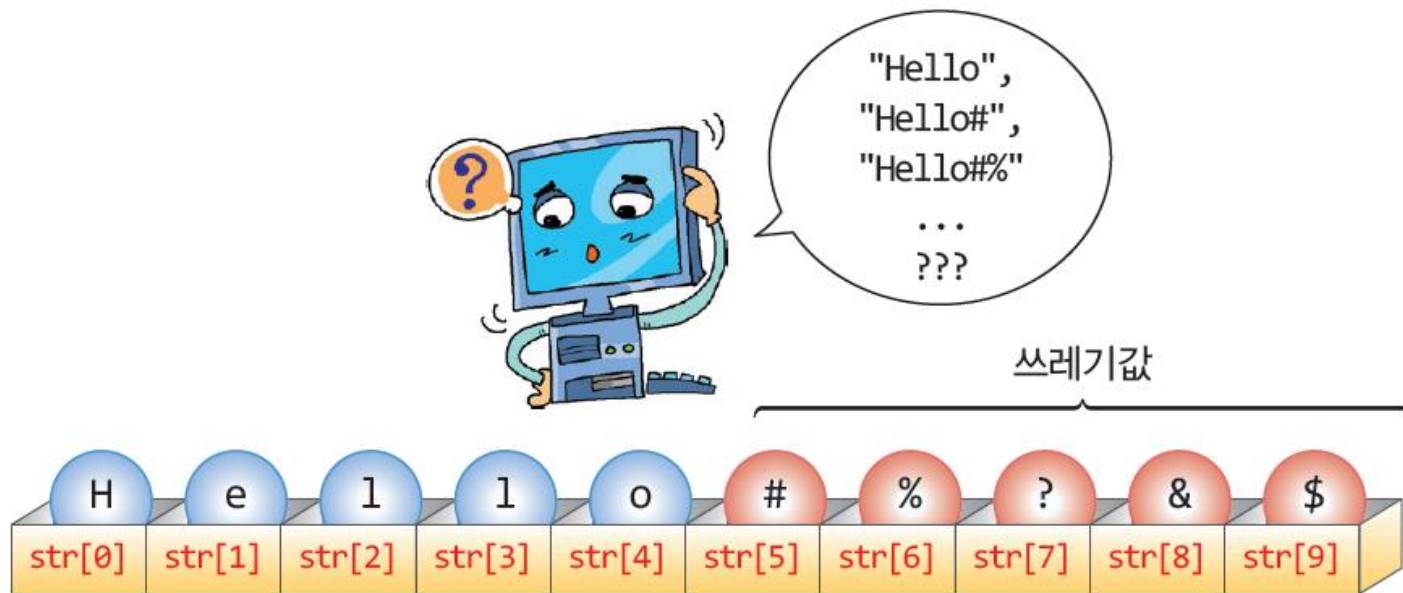


그림 10.3 NULL 문자의 필요성: 정상적인 데이터와 쓰레기 값을 분리하기 위해서이다.

문자열 배열의 초기화

```
char str[6] = { 'H', 'e', 'l', 'l', 'o', '\0' };    // 배열의 초기화 방식
```

```
char str[6] = "Hello";                            // 문자열로 초기화
```

```
char str[] = "Hello";                             // 배열의 크기는 6
```

예제 #1

```
#include <stdio.h>

int main(void)
{
    char str1[6] = "Seoul";
    char str2[3] = { 'i', 's', '\0' };
    char str3[] = "the capital city of Korea.";

    printf("%s %s %s\n", str1, str2, str3);
    return 0;
}
```

C:\ Microsoft Visual Studio 디버그 콘솔

Seoul is the capital city of Korea.

예제 #2

```
#include <stdio.h>

int main(void)
{
    char str[] = "komputer";

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

    return 0;
}
```

C:\ Microsoft Visual Studio 디버그 콘솔

komputer
computer

예제 #3

```
#include <stdio.h>

int main(void)
{
    char str[] = "A barking dog never bites";
    int i = 0;

    while (str[i] != 0)
        i++;
    printf("문자열 %s의 길이는 %d입니다.\n", str, i);

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

문자열 A barking dog never bites의 길이는 25입니다.

중간점검

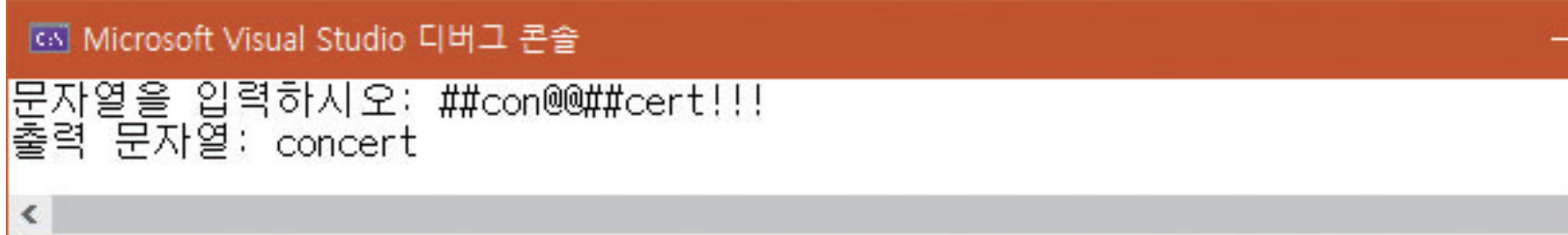


중간점검

1. C에서 문자열은 어떻게 정의되는가?
2. 문자열에서 **NULL** 문자의 역할은 무엇인가?
3. **NULL** 문자의 아스키 코드 값은 얼마인가?
4. **NULL** 문자로 끝나지 않는 문자열을 출력하면 어떻게 되는가?
5. **B**, **'B'**, **"B"**의 차이점을 설명하라.
6. 변경 가능한 문자열은 어디에 저장되는가?
7. 문자열의 크기보다 문자 배열의 크기를 하나 더 크게 하는 이유는 무엇인가?
8. 문자 배열을 문자열로 초기화하는 방법을 아는 대로 설명하라.

Lab: 문자열 전처리하기

- 사용자로부터 문자열을 받아서 특수 문자들을 제거하고 알파벳 문자만 남기는 프로그램을 작성해보자.



```
C:\> Microsoft Visual Studio 디버그 콘솔
문자열을 입력하시오: ##con@@##cert!!!
출력 문자열: concert
```

The screenshot shows a debug console window with an orange header bar. The text inside shows a prompt to enter a string, followed by the input '##con@@##cert!!!' and the output 'concert' after processing. A scrollbar is visible at the bottom.

Lab: 문자열 전처리하기

```
#include <stdio.h>

int main(void) {
    char line[100] = "";
    char line2[100] = "";

    printf("문자열을 입력하시오: ");
    gets_s(line, sizeof(line));          // 한 줄의 문자열 입력

    int k = 0;
    for (int i = 0; line[i] != '\0'; ++i) {
        while (line[i] >= 'A' && line[i] <= 'z') {
            line2[k++] = line[i++];
        }
    }
    line2[k] = '\0';
    printf("출력 문자열: %s\n", line2);
    return 0;
}
```

문자 입출력

함수	설명
getchar()	하나의 문자를 받아서 반환한다. 버퍼를 사용한다.
putchar()	하나의 문자를 받아서 출력한다.
_getch()	하나의 문자를 받아서 반환한다. 버퍼를 사용하지 않는다.
_putch()	하나의 문자를 받아서 출력한다.

#include <conio.h> 헤더가 필요함
리눅스나 Mac OS에서는 사용 불가
Visual studio에서만 사용 가능

예제

```
#include <stdio.h>

int main(void)
{
    int ch; // 정수형에 주의
    ch = getchar(); // 첫 번째 문자를 입력받는다.
    putchar(ch); // 문자를 출력한다.

    return 0;
}
```



The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. It has a dark red header bar with the text 'Microsoft Visual Studio 디버그 콘솔' and a small icon on the left. Below the header, the console area is white and contains two lines of text, both 'a'. A horizontal scrollbar is visible at the bottom of the console area.

예제

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

int main(void)
{
    int ch; // 정수형에 주의
    ch = _getch(); // 첫 번째 문자를 입력받는다.
    _putch(ch); // 문자를 출력한다.

    return 0;
}
```



Lab: 사용자로부터 패스워드 입력받기

- 흔히 패스워드를 입력받을 때는 사용자가 입력한 문자 대신에 "*" 문자를 표시한다.
- `_getch()`를 이용하여 사용자가 입력한 문자를 비밀스럽게 입력받는 프로그램을 작성해보자. 패스워드는 8자리라고 가정하자

C# Microsoft Visual Studio 디버그 콘솔

패스워드를 입력하시오: *****

입력된 패스워드는 다음과 같습니다: 12345678

<

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

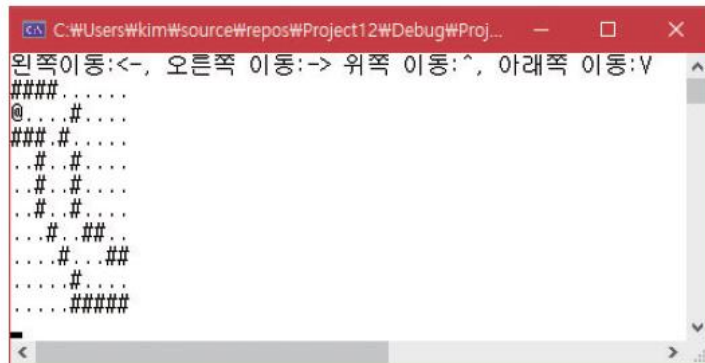
int main(void)
{
    char password[10];
    int i;

    system("cls");
    printf("패스워드를 입력하시오: ");
    for (i = 0; i < 8; i++) {
        password[i] = _getch();
        printf("*");
    }
    password[i] = '\0';
    printf("\n");

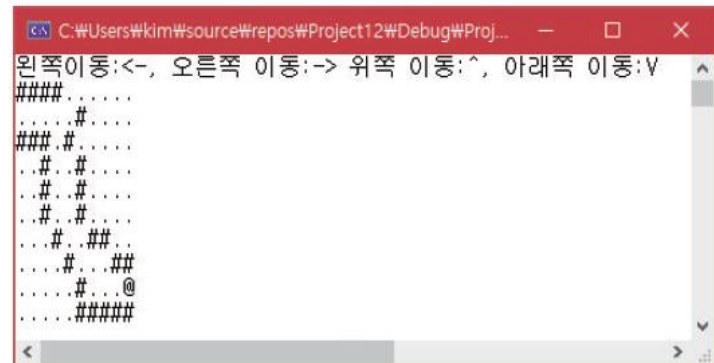
    printf("입력된 패스워드는 다음과 같습니다: ");
    printf("%s\n", password);
    _getch();
    return 0;
}
```

Lab: 화살표 키 입력받기

- 화살표 키를 `_getch()`로 입력받아서 주인공 `@`을 왼쪽 입구에서 오른쪽 출구로 이동시키는 게임을 작성해보자. 한 번이라도 벽에 닿으면 실패이다.



->



```
int ch = _getch();  
if (ch == 224) {
```

```
    int ch2 = _getch();  
    if (ch2 == 75) xpos--;  
    else if (ch2 == 80) ypos++;  
    else if (ch2 == 72) ypos--;  
    else if (ch2 == 77) xpos++;
```

```
}
```

화살표 키 처리

Sol:

```
#include <stdio.h>
#include <conio.h>
int main(void) {
    char board[10][10] = { {'#', '#', '#', '#', '.', '.', '.', '.', '.', '.'},
        {'.', '.', '.', '.', '.', '#', '.', '.', '.', '.'},
        {'#', '#', '#', '.', '#', '.', '.', '.', '.', '.'},
        {'.', '.', '#', '.', '.', '#', '.', '.', '.', '.'},
        {'.', '.', '#', '.', '.', '#', '.', '.', '.', '.'},
        {'.', '.', '#', '.', '.', '#', '.', '.', '.', '.'},
        {'.', '.', '.', '#', '.', '.', '#', '#', '.', '.'},
        {'.', '.', '.', '.', '#', '.', '.', '.', '#', '#'},
        {'.', '.', '.', '.', '.', '#', '.', '.', '.', '.'},
        {'.', '.', '.', '.', '.', '#', '#', '#', '#', '#'} };

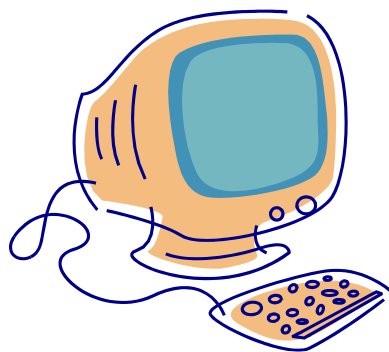
    int xpos = 0, ypos = 1;
    board[ypos][xpos] = '@';
```

// 사용자로부터 위치를 받아서 보드에 표시한다.

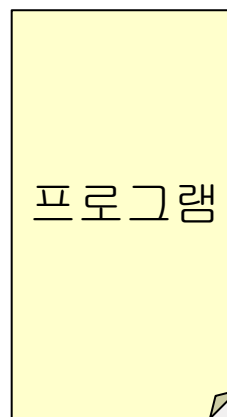
```
while (1) {
    system("cls");
    printf("왼쪽이동:<-, 오른쪽 이동:-> 위쪽 이동:^, 아래쪽 이동:\n");
    for (int y = 0; y < 10; y++) {
        for (int x = 0; x < 10; x++) printf("%c", board[y][x]);
        printf("\n");
    }
    board[ypos][xpos] = '.';
    int ch = _getch();
    if (ch == 224) {
        int ch2 = _getch();
        if (ch2 == 75) xpos--;
        else if (ch2 == 80) ypos++;
        else if (ch2 == 72) ypos--;
        else if (ch2 == 77) xpos++;
    }
    board[ypos][xpos] = '@';
}
return 0;
}
```

문자열 입출력 라이브러리 함수

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

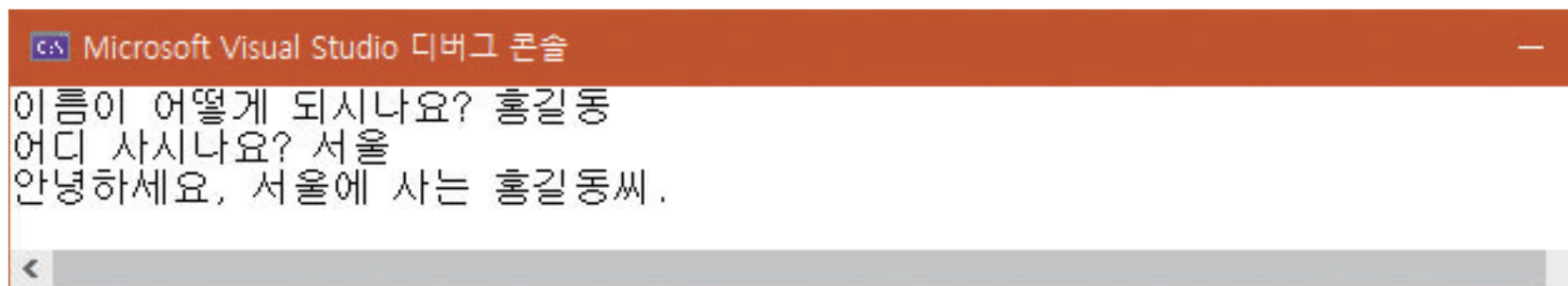


...Hello World!...



예제 #1

- 사용자의 이름을 받아서 친근하게 대답하는 프로그램을 작성해보자.



```
Microsoft Visual Studio 디버그 콘솔
이름이 어떻게 되시나요? 홍길동
어디 사시나요? 서울
안녕하세요, 서울에 사는 홍길동씨.
```

예제 #1

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

int main(void)
{
    char name[100];
    char address[100];

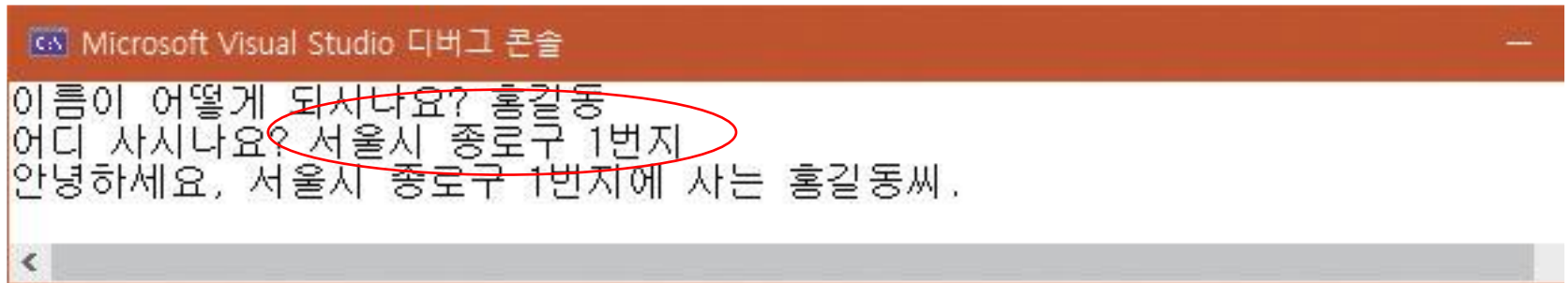
    printf("이름이 어떻게 되시나요? ");
    scanf("%s", name);

    printf("어디 사시나요? ");
    scanf("%s", address);

    printf("안녕하세요, %s에 사는 %s씨.\n", address, name);
    return 0;
}
```

예제 #2

- 사용자의 이름을 받아서 친근하게 대답하는 프로그램을 작성해보자.



The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. It contains the following text: '이름이 어떻게 되시나요? 홍길동' (What is your name? Honggildong), '어디 사시나요? 서울시 종로구 1번지' (Where do you live? 1st floor, Jongno-gu, Seoul), and '안녕하세요, 서울시 종로구 1번지에 사는 홍길동씨.' (Hello, Mr. Honggildong who lives in 1st floor, Jongno-gu, Seoul). A red oval highlights the input text '홍길동' and '서울시 종로구 1번지'.

```
gets_s(char buffer[] , int size);
```

문자 배열

최대 문자 개수

예제 #2

```
#include <stdio.h>

int main(void)
{
    char name[100];
    char address[100];

    printf("이름이 어떻게 되시나요? ");
    gets_s(name, 99);

    printf("어디 사시나요? ");
    gets_s(address, 99);

    printf("안녕하세요, %s에 사는 %s씨.\n", address, name);
    return 0;
}
```

중간 점검



경고

이전에 사용하던 `gets()` 함수는 안전성 문제로 최근의 C언어 표준에서 완전히 금지되었다. 되도록 사용하지 않도록 하자.



중간점검

1. 한 줄의 텍스트를 입력받는 문장을 작성하라.
2. 사용자로부터 하나의 단어를 입력받는 문장을 작성하라.



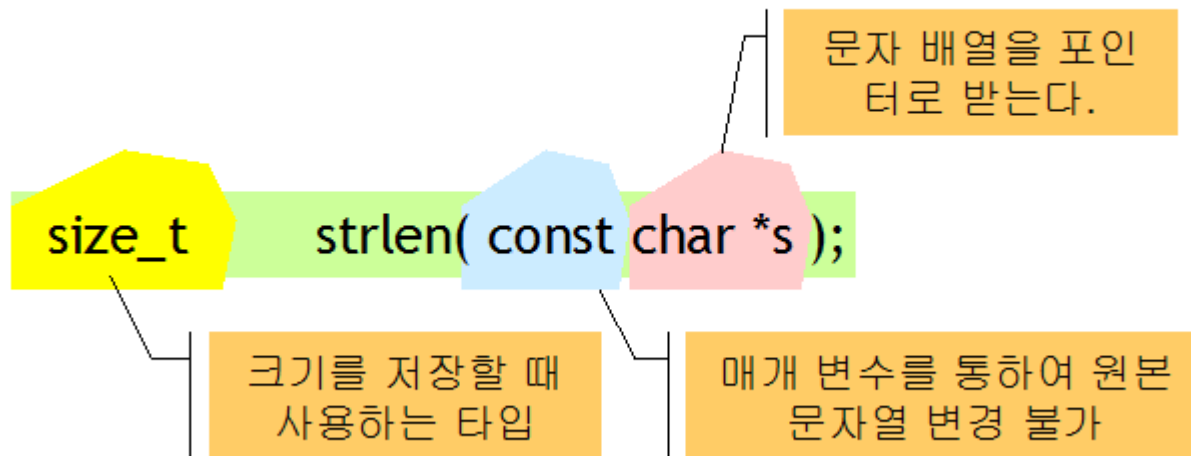
문자열 처리 라이브러리

- 문자열을 사용하다보면 두 개의 문자열을 붙이는 작업이나 두 개의 문자열을 서로 비교하는 작업들이 필요해진다.
- 문자열 함수들은 `string.h`에 선언되어 있다. 따라서 이들 함수들을 사용하려면 `string.h`를 프로그램의 첫 부분에서 포함시켜야 한다.

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

함수	설명
<code>strlen(s)</code>	문자열 <code>s</code> 의 길이를 구한다.
<code>strcpy(s1, s2)</code>	<code>s2</code> 를 <code>s1</code> 에 복사한다.
<code>strcat(s1, s2)</code>	<code>s2</code> 를 <code>s1</code> 의 끝에 붙여넣는다.
<code>strcmp(s1, s2)</code>	<code>s1</code> 과 <code>s2</code> 를 비교한다.

문자열 길이: strlen()



문자열 길이: strlen()

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char s[] = "abcdefgh";
    int len = strlen(s);
    printf("문자열 %s의 길이는%d \n", s, len);
    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

문자열 abcdefgh의 길이는8

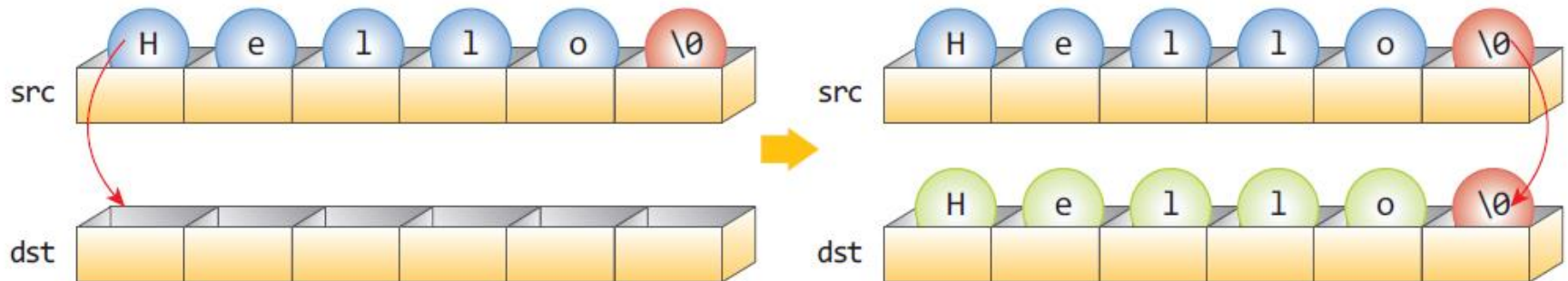
문자열 복사: strcpy()

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
int main(void)
{
    char src[] = "Hello";
    char dst[6];
    strcpy(dst, src);
    printf("복사된 문자열 = %s \n", dst);
    return 0;
}
```



The screenshot shows the Microsoft Visual Studio debug console. The title bar at the top reads "Microsoft Visual Studio 디버그 콘솔". The console output displays the text "복사된 문자열 = Hello". A horizontal scrollbar is visible at the bottom of the console window.

문자열 복사: strcpy()

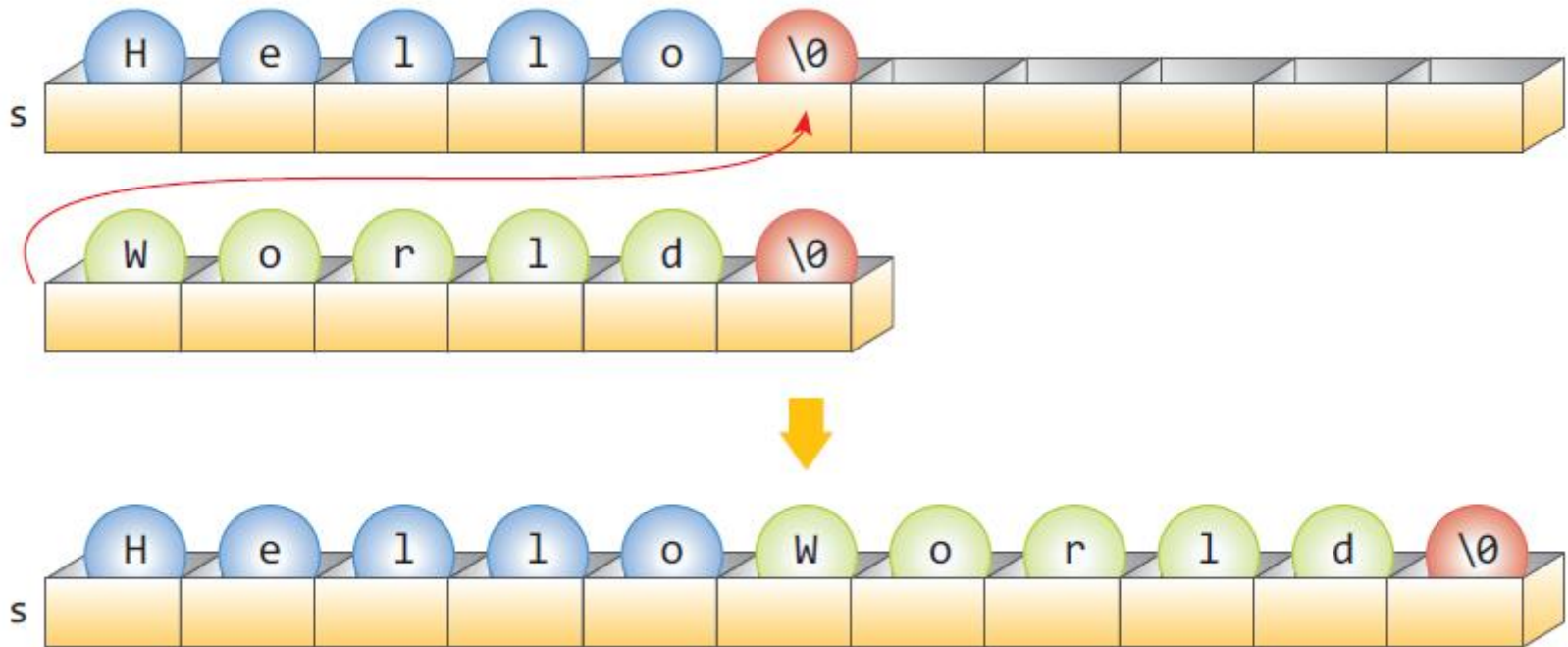


문자열 연결

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
int main(void)
{
    char s[11] = "Hello";
    strcat(s, "World"); // s에 문자열 "World"를 붙인다.
    printf("%s \n", s);
    return 0;
}
```



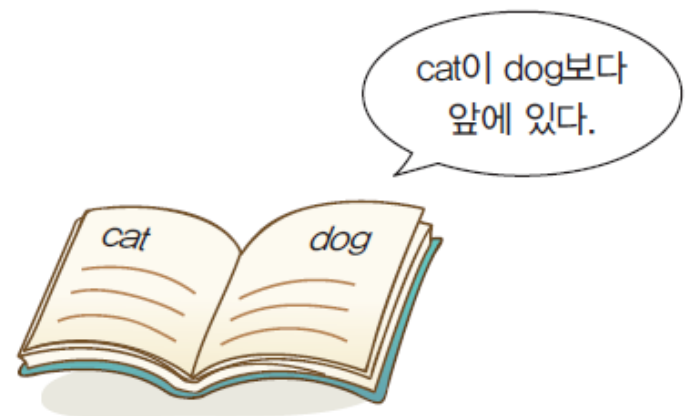
문자열 연결



문자열 비교

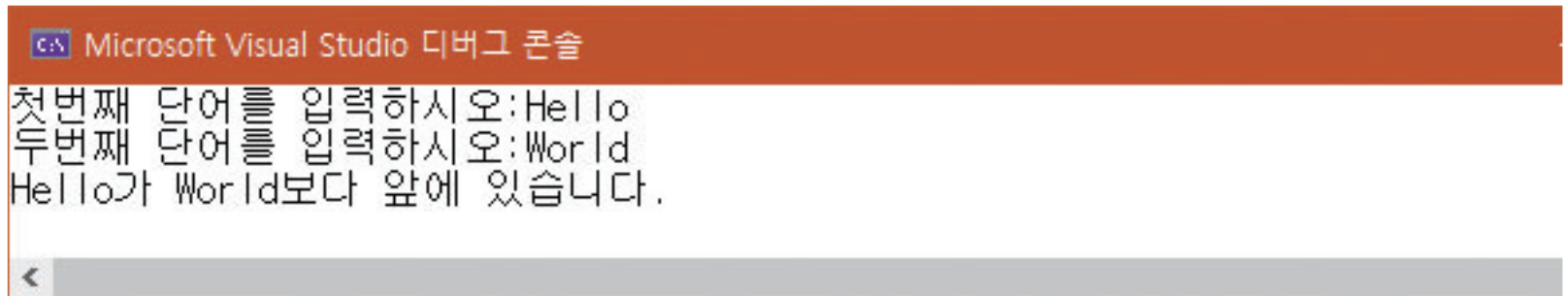
```
int strcmp( const char *s1, const char *s2 );
```

반환값	s1과 s2의 관계
<0	s1이 s2보다 앞에 있다.
0	s1이 s2와 같다.
>0	s1이 s2보다 뒤에 있다.



예제

- 사용자로부터 문자열 2개를 받아서 비교하는 프로그램을 작성해보자.



```
C:\> Microsoft Visual Studio 디버그 콘솔  
첫번째 단어를 입력하시오:Hello  
두번째 단어를 입력하시오:World  
Hello가 World보다 앞에 있습니다.
```

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

int main(void)
{
    char s1[80]; // 첫 번째 단어를 저장할 문자배열
    char s2[80]; // 두 번째 단어를 저장할 문자배열
    int result;

    printf("첫 번째 단어를 입력하시오:");
    scanf("%s", s1, sizeof(s1));
    printf("두 번째 단어를 입력하시오:");
    scanf("%s", s2, sizeof(s2));

    result = strcmp(s1, s2);
    if (result < 0)
        printf("%s가 %s보다 앞에 있습니다.\n", s1, s2);
    else if (result == 0)
        printf("%s가 %s와 같습니다.\n", s1, s2);
    else
        printf("%s가 %s보다 뒤에 있습니다.\n", s1, s2);
    return 0;
}
```

문자열을 수치로 변환: atoi, sscanf, sprintf

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    const char s[] = "100";
    char t[100] = "";
    int i;
    printf("%d \n", atoi("100")); // atoi()는 문자열 "100"을 정수 100으로 바꾼다.
    sscanf(s, "%d", &i);           // 문자열 "100"을 정수 100으로 i에 저장.
    sprintf(t, "%d", 100);         // 정수 100을 문자열 "100"으로 t에 저장
    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

100

참고사항



Q 문자열을 비교할 때 다음과 같이 하면 안되는지?

```
char *s;  
if( s == "langauge" )
```

A C언어는 문자들의 배열로 문자열을 나타낸다. C에서는 배열을 다른 배열과 연산자로 비교할 수는 없다. 두 개의 문자열을 비교하려면 반드시 `strcmp()`를 사용하여야 한다.

```
if( strcmp(s, "language") == 0 )
```

참고사항



Q 다음 문장이 오류를 발생하는 이유는 무엇인가?

```
char s[100] ;  
s = "A friend in power is a friend lost.";
```

A C에서는 초기화 문장을 제외하고, 배열에 문자열을 직접 대입할 수는 없다. 반드시 `strcpy()` 함수를 사용하여야 한다.

```
strcpy(s, "A friend in power is a friend lost.");
```

중간점검



중간점검

1. `sprintf()`에 대하여 더 조사해보자. 어떤 때 사용하는 함수인가?
2. `sscanf()`에 대하여 더 조사해보자. 어떤 때 사용하는 함수인가?

Lab: 단답형 퀴즈 채점

- 단답형 문제를 채점하는 프로그램을 작성해보자. 사용자가 정답을 입력할 때까지 반복한다. 단답형 문제이기 때문에 문자열이 일치하는지를 검사한다

```
C:\> Microsoft Visual Studio 디버그 콘솔  
임베디드 장치에 가장 많이 사용되는 언어는? Java  
임베디드 장치에 가장 많이 사용되는 언어는? Python  
임베디드 장치에 가장 많이 사용되는 언어는? C  
맞았습니다!
```



Sol:

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

int main(void)
{
    char key[] = "C";
    char buffer[80] = "";

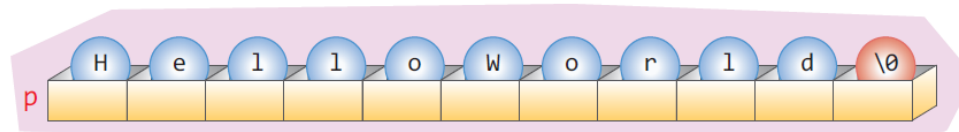
    do {
        printf("임베디드 장치에 가장 많이 사용되는 언어는? ");
        gets_s(buffer, sizeof(buffer));
    } while (strcmp(key, buffer) != 0);

    printf("맞았습니다!");
    return 0;
}
```

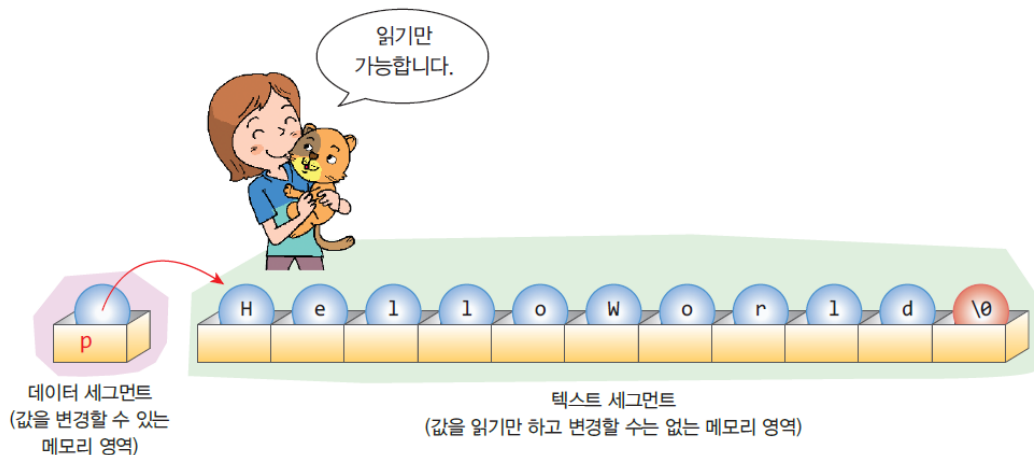
문자열과 포인터

- 아래의 2 문장은 어떻게 다를까?

```
char s[] = "HelloWorld"; // ①  
char *p = "HelloWorld"; // 읽기전용 메모리에 문자열 저장
```

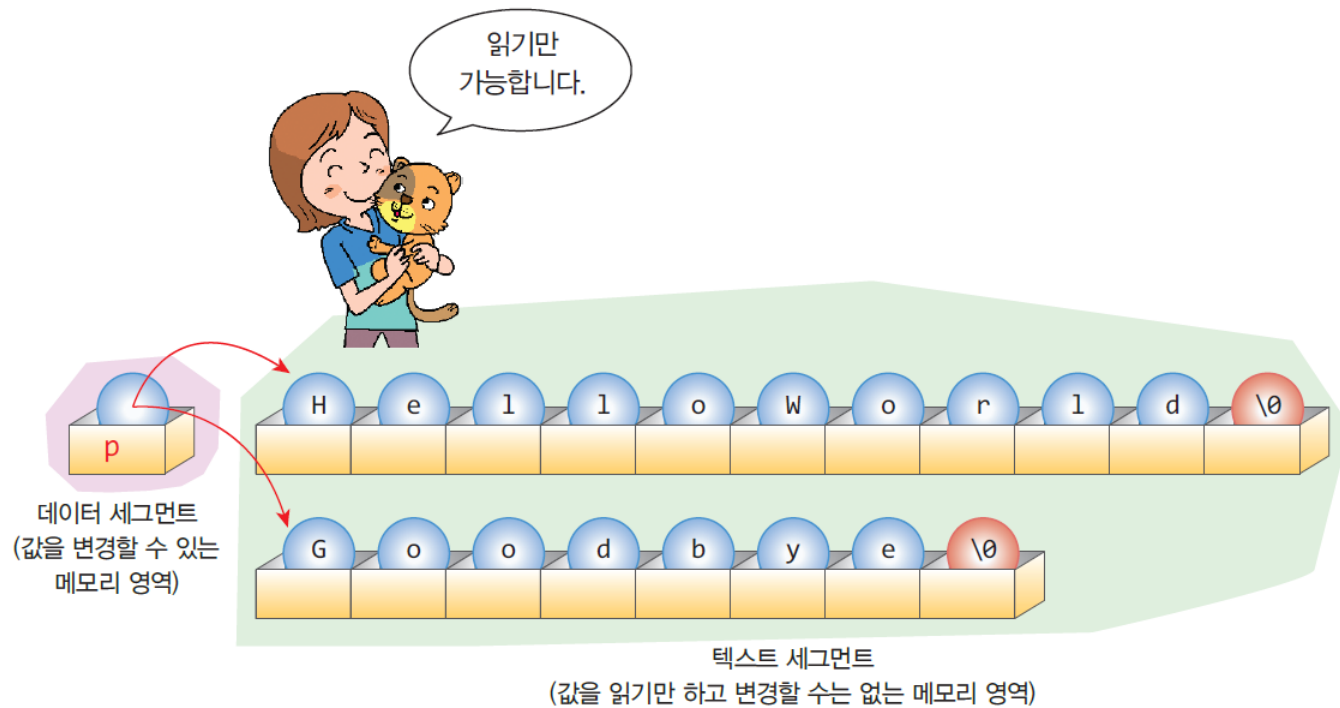


데이터 세그먼트
(값을 변경할 수 있는 메모리 영역)



문자열과 포인터

```
char *p = "HelloWorld";  
p = "Goodbye";
```



예제

```
#include <stdio.h>

int main(void)
{
    char s[] = "HelloWorld";
    char* p = "HelloWorld";

    s[0] = 'h'; // 변경 가능하다.
    // p[0] = 'h'; // 이 문장을 실행하면 프로그램의 작동이 중지된다.

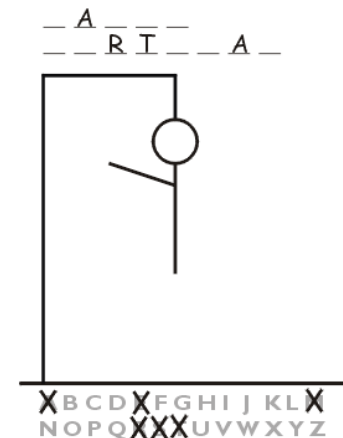
    printf("포인터가 가리키는 문자열 = %s \n", p);
    p = "Goodbye"; // 이것은 가능하다. 큰 장점이다.
    printf("포인터가 가리키는 문자열 = %s \n", p);
    return 0;
}
```

C:\> Microsoft Visual Studio 디버그 콘솔

```
포인터가 가리키는 문자열 = HelloWorld
포인터가 가리키는 문자열 = Goodbye
```


Lab: 행맨

- 빈칸으로 구성된 문자열이 주어지고 사용자는 문자열에 들어갈 글자들을 하나씩 추측해서 맞추는 게임



So

```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

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

교과서엔 빠짐

```
int main(void) {
```

```
    char solution[100] = "meet at midnight";           // 정답이 들어 있는
```

문자 배열

```
    char answer[100] = "____ _"; // 현재까지 사용자가 맞춘
```

문자열

```
    char ch;
```

```
    int i;
```

```
    while (1) {
```

```
        printf("\n문자열을 입력하시오: %s \n", answer);
```

```
        printf("글자를 추측하시오: ");
```

```
        ch = _getch();
```

```
        // 사용자가 입력한 문자를 answer[]에 넣어서 정답을 비교한다.
```

```
        for (i = 0; solution[i] != '\0'; i++) {
```

```
            // 사용자가 맞추었으면 글자를 보이게 한다.
```

```
            if (solution[i] == ch)
```

```
                answer[i] = ch;
```

```
        }
```

```
        if (strcmp(solution, answer) == 0) break; // 정답과
```

일치하는지를 검사

```
    }
```

```
    return 0;
```

```
}
```

도전문제



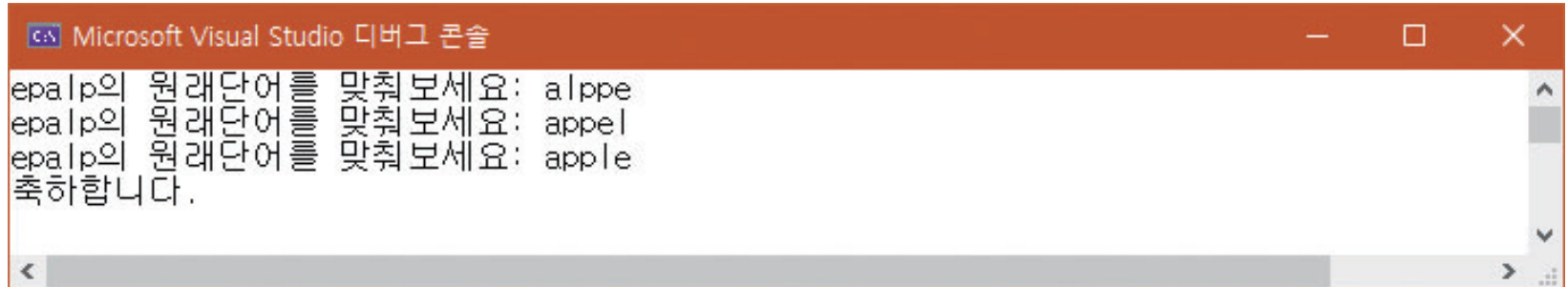
도전문제

- (1) 여러 개의 문자열 중에서, 하나를 정답으로 선택하도록 위의 프로그램을 업그레이드하여 보자.
- (2) solution 문자열에서 "_____" 문자열을 자동으로 생성해보자.



Lab: 단어 애나그램 게임

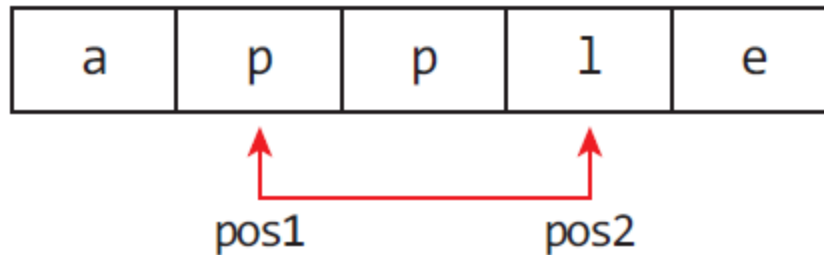
- 영어 단어를 이루는 글자들이 뒤죽박죽 섞인 것을 받아서 순서대로 재배치하는 게임을 애나그램 게임이라고 한다.



```
Microsoft Visual Studio 디버그 콘솔
epa|p의 원래단어를 맞춰보세요: alppe
epa|p의 원래단어를 맞춰보세요: appel
epa|p의 원래단어를 맞춰보세요: apple
축하합니다.
```

Lab: 단어 애나그램 게임

- 문자열 안의 글자들을 섞으려면 난수가 필요하다. 두 개의 난수를 발생시켜서 그 위치의 글자들을 서로 바꾸면 된다. 이것을 문자열의 길이만큼 반복한다. 물론 난수의 범위는 문자열안이어야 한다.



Sc

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include <stdlib.h>
```

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

```
#define SOL "apple"
```

```
int main(void)
```

```
{
```

```
    char s[100] = SOL;
```

```
    char ans[100];
```

```
    int i, len;
```

```
    len = strlen(s);
```

```
    for (i = 0; i < len; i++) {
```

```
        int pos1 = rand() % len;
```

```
        int pos2 = rand() % len;
```

```
        char tmp = s[pos1];
```

```
        s[pos1] = s[pos2];
```

```
        s[pos2] = tmp;
```

```
    }
```

```
    do {
```

```
        printf("%s의 원래단어를 맞춰보세요: ", s);
```

```
        scanf("%s", ans);
```

```
    } while (strcmp(ans, SOL) != 0);
```

```
    printf("축하합니다. \n");
```

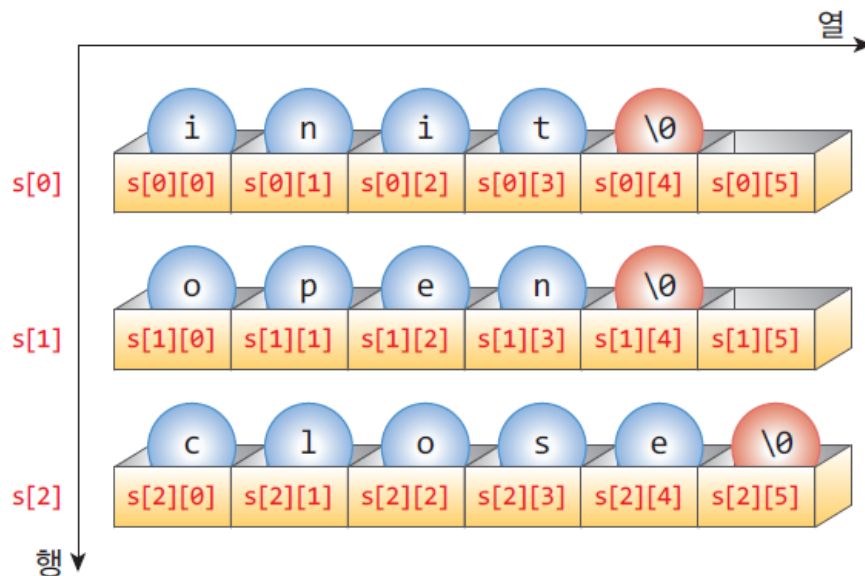
```
    return 0;
```

```
}
```

교과서 오타

문자열의 배열: 2차원 문자형 배열 사용

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



예제

- 다음과 같은 메뉴를 2차원 문자 배열에 저장하고 화면에 출력하는 프로그램을 작성해보자.



```
Microsoft Visual Studio 디버그 콘솔
0 번째 메뉴: init
1 번째 메뉴: open
2 번째 메뉴: close
3 번째 메뉴: read
4 번째 메뉴: write
```

The image shows a screenshot of the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. The title bar is orange and contains the text 'Microsoft Visual Studio 디버그 콘솔' along with standard window control buttons (minimize, maximize, close). The main area of the console is white and displays five lines of text, each representing a menu item: '0 번째 메뉴: init', '1 번째 메뉴: open', '2 번째 메뉴: close', '3 번째 메뉴: read', and '4 번째 메뉴: write'. The text is in a monospaced font. On the right side of the console, there is a vertical scrollbar with up and down arrows. At the bottom, there is a horizontal scrollbar with left and right arrows.

예제

```
#include <stdio.h>

int main(void)
{
    int i;
    char menu[5][10] = {
        "init",
        "open",
        "close",
        "read",
        "write"
    };

    for (i = 0; i < 5; i++)
        printf("%d 번째 메뉴: %s \n", i, menu[i]);

    return 0;
}
```

중간점검



중간점검

1. "C", "JAVA", "C++", "BASIC" 등을 저장할 수 있는 방법을 가능한 한 많이 제시하라.
2. 2차원 문자 배열 `s[][]`에서 첫 번째 문자열을 `printf()`를 이용하여 화면에 출력하는 문장을 작성하라.

Lab: 한영 사전의 구현

- 3차원 문자 배열을 이용하여 간단한 한영 사전을 구현하여 보자.
영어 단어가 `dic[i][0]`에 저장된다. 한글 설명은 `dic[i][1]`에 저장된다.



```
Microsoft Visual Studio 디버그 콘솔
단어를 입력하시오:rain
rain: 비
```

The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. It contains two lines of text: '단어를 입력하시오:rain' (Enter word: rain) and 'rain: 비' (rain: rain). The window has a standard Windows title bar with minimize, maximize, and close buttons.

Sol:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#define WORDS 5

int main(void)
{
    int i, index;
    char dic[WORDS][2][30] = {
        { "book", "책" },
        { "boy", "소년" },
        { "computer", "컴퓨터" },
        { "language", "언어" },
        { "rain", "비" },
    };
    char word[30];
```

Sol:

```
printf("단어를 입력하시오:");  
scanf("%s", word);  
index = 0;  
for (i = 0; i < WORDS; i++)  
{  
    if (strcmp(dic[index][0], word) == 0)  
    {  
        printf("%s: %s\n", word, dic[index][1]);  
        return 0;  
    }  
    index++;  
}  
printf("사전에서 발견되지 않았습니다.\n");  
  
return 0;  
}
```

Lab: 문자열의 정렬

- 문자열들을 알파벳 순으로 정렬할 수 있을까?

C# Microsoft Visual Studio 디버그 콘솔

```
apple  
avocado  
banana  
pear  
pineapple  
tomato
```



Sol:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define SIZE 6
int main(void)
{
    int i, k;
    char fruits[SIZE][20] = {
        "pineapple",
        "banana",
        "apple",
        "tomato",
        "pear",
        "avocado"
    };
};
```

Sol:

```
for (k = 0; k < SIZE; k++) {  
    for (i = 0; i < SIZE - 1; i++) {  
        if (strcmp(fruits[i], fruits[i + 1]) > 0) {  
            char tmp[20];  
            strcpy(tmp, fruits[i]);  
            strcpy(fruits[i], fruits[i + 1]);  
            strcpy(fruits[i + 1], tmp);  
        }  
    }  
}  
for (k = 0; k < SIZE; k++)  
    printf("%s \n", fruits[k]);  
return 0;  
}
```


Mini Project: 텍스트 암호화

- 이번 미니 프로젝트에서는 주어진 텍스트를 암호화시켜보자. 사용자는 암호화 알고리즘을 마음대로 선택할 수 있다.
- 권장하는 암호화 방법은 XOR 암호화 방법이다. 이 알고리즘에서는 암호키와 문자열의 모든 문자에 대하여 비트 XOR 연산자를 적용하여 문자열을 암호화한다.
- 출력을 해독하려면 동일한 키를 사용하여 XOR 함수를 다시 적용하기 만하면 암호가 제거된다.



Mini Project: 텍스트 압축

- 간단한 텍스트 압축 프로그램을 작성해본다. 어떤 압축 방법을 선택해도 좋다. 텍스트는 사용자로부터 직접 입력된다. 알파벳 영문자로 한정한다.
- 권장하는 방법은 런길이 엔코딩(run length encoding)이다.
- 런길이 부호화는 무손실 압축 방법 중 하나로서, 텍스트를 압축할 때 반복되는 문자가 있으면, 이것을 반복되는 개수와 반복되는 문자로 바꾸는 방법이다. 예를 들어서 "wwwssssssssschh"는 "3w9s1c2h"로 압축하는 것이다.

