

Basic Computer Programming

Lecture 10

Electrical & Electronics Engineering
Chung-Ang University

Contents

- See how the string is represented inside the computer.
- String input/output functions will be used.
- How to compare and copy strings.
- Multiple strings can be stored and processed.

The programs we will make in this chapter

- Let's write a "Hangman" game.



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

문자열을 입력하시오: _ _ _ _ _
자리를 추측하시오:
문자열을 입력하시오: _ _ _ a _ _ _ _ _
자리를 추측하시오:
문자열을 입력하시오: _ee_ a _ _ _ _ _
자리를 추측하시오: ■
```

- Let's write an English word anagram game.



```
C:\> Microsoft Visual Studio 디버그 콘솔

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

String

- A string is a collection of characters, such as "Hello".

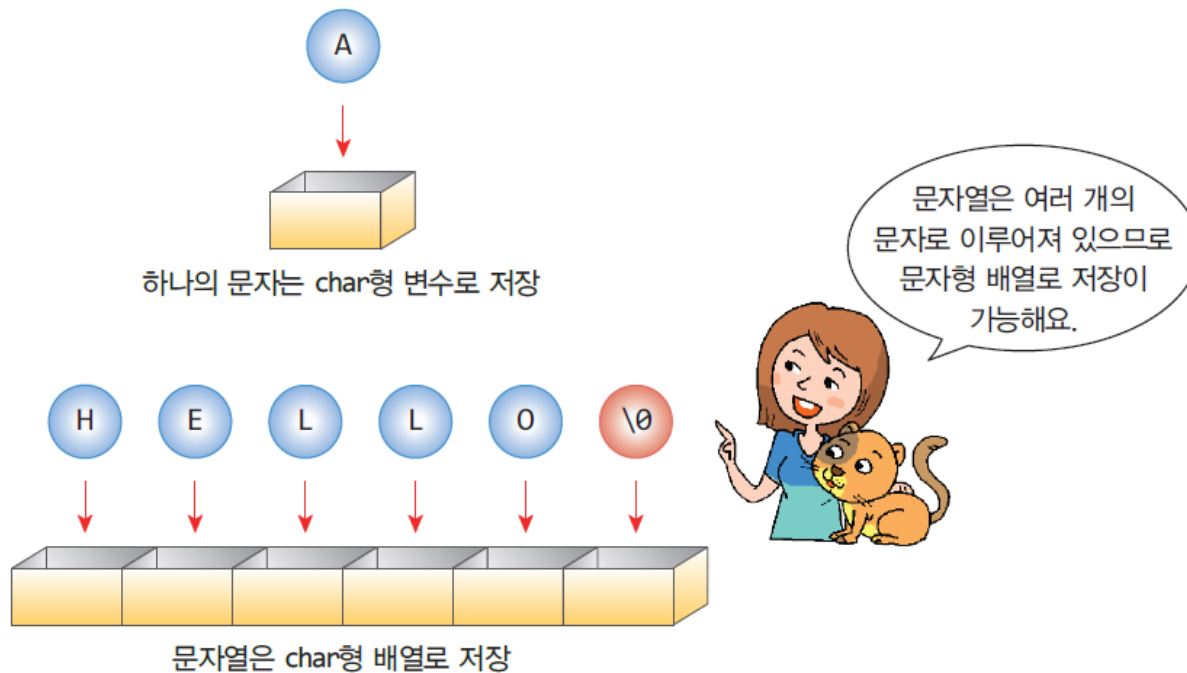
"A"

"Hello"

"Mr. Hong"

String representation method

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



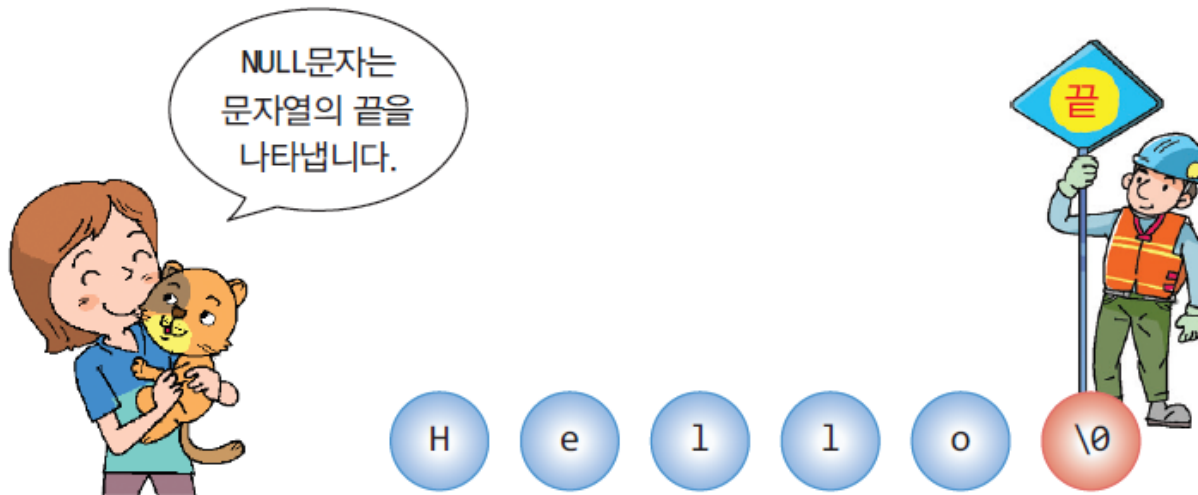
How to Express Characters

- On a computer, each character is displayed with a numeric code.
- ASCII code: standard 8-bit character code
 - Expressing characters using numbers from 0 to 127
- The C language uses only the ASCII code.

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 characters

- The end of the string must end with a special value called the null character.



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

Why NULL characters are needed?

- Human knows that "Hello" is the only meaningful string and "#%?&\$" is a trash character, but the computer does not know where it is meaningful.

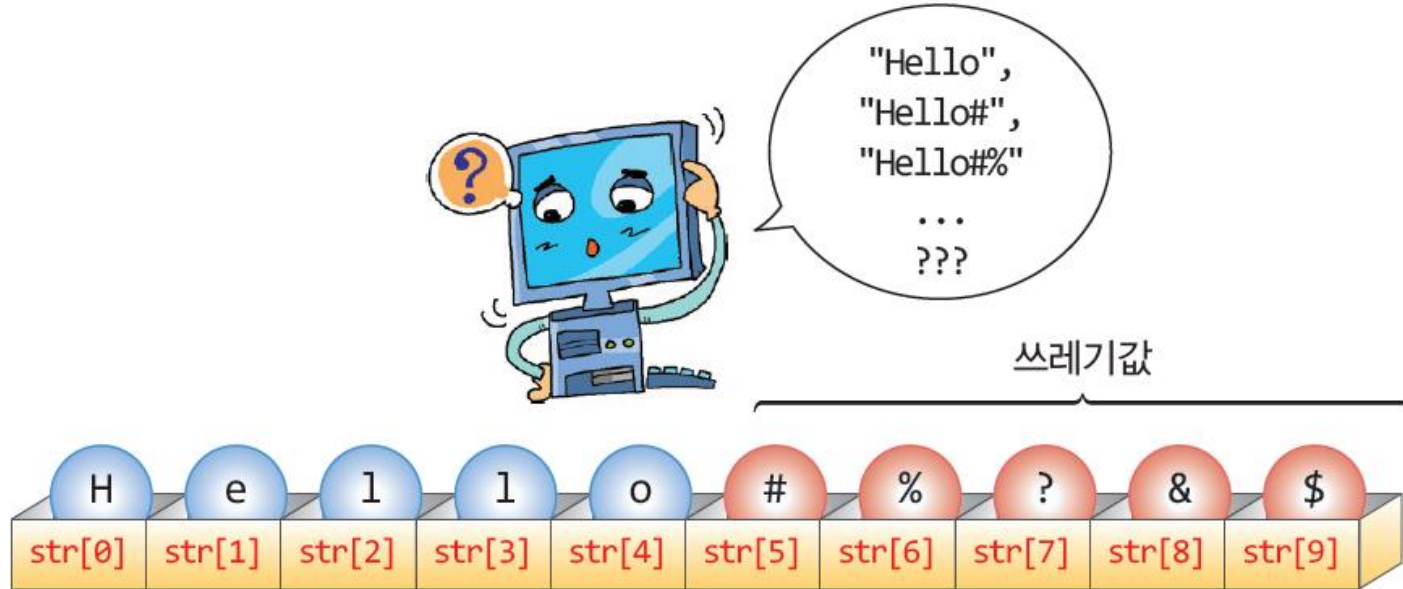


그림 10.3 NULL 문자의 필요성: 정상적인 데이터와 쓰레기 값을 분리하기 위해서이다.

Initialize string array

```
char str[6] = { 'H', 'e', 'l', 'l', 'o', '\0' };    // How the array is initialized
```

```
char str[6] = "Hello";                            // Initialize to String
```

```
char str[] = "Hello";                             // Array size is 6
```

Example #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.

Example #2

```
#include <stdio.h>

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

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

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

komputer
computer

Example #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.

Interim check

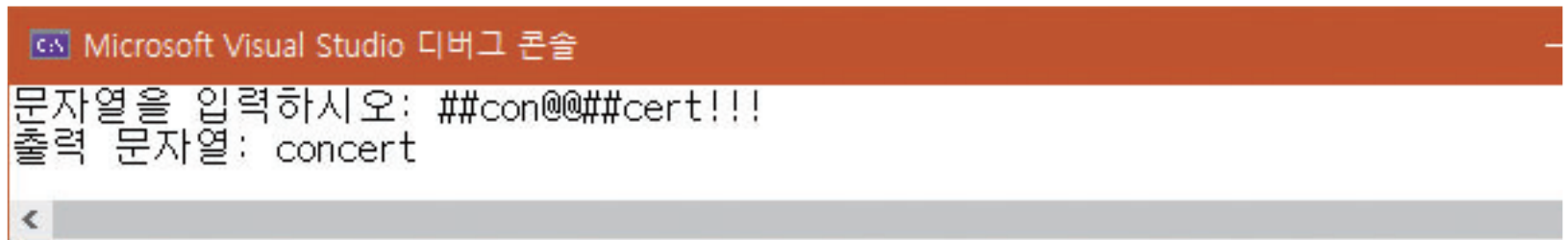


중간점검

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: Preprocess String

- Let's write a program that takes strings from users and removes special characters and leaves only alphabetic characters.



The screenshot shows a debug console window from Microsoft Visual Studio. The title bar is orange and contains the text "C:\> Microsoft Visual Studio 디버그 콘솔". The console area has a white background and displays two lines of Korean text. The first line is "문자열을 입력하시오: ##con@@##cert!!!" and the second line is "출력 문자열: concert". A scrollbar is visible on the right side of the console area.

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

Example #3

```
#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;
}
```

String input and output

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.

Example

```
#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 a debug console window from Microsoft Visual Studio. The title bar is dark red and contains the text "Microsoft Visual Studio 디버그 콘솔" (Microsoft Visual Studio Debug Console). The console area has a white background and shows two lines of output, both of which are the letter 'a'. A horizontal scrollbar is visible at the bottom of the console area.

Example

```
#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: Get password input from user

- When a password is entered, a "*" character is displayed instead of the character entered by the user.
- Let's write a program that secretly receives characters entered by users using `_getch()`. Suppose the password is eight digits



```
C:\> Microsoft Visual Studio 디버그 콘솔  
패스워드를 입력하시오: *****  
입력된 패스워드는 다음과 같습니다: 12345678
```

The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar is orange and contains the text "C:\> Microsoft Visual Studio 디버그 콘솔". The console area has a white background and displays two lines of text in Korean. The first line is "패스워드를 입력하시오: *****" (Enter password: *****). The second line is "입력된 패스워드는 다음과 같습니다: 12345678" (The entered password is as follows: 12345678). Below the text is a horizontal scrollbar with a grey track and a white slider.

Sol.

```
#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;
}
```

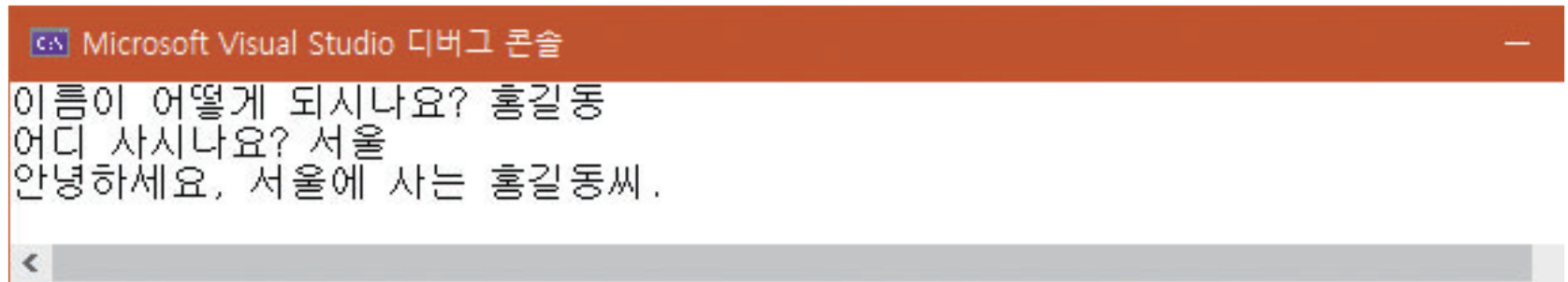
String I/O library functions

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.



Example #1

- Let's write a program that receives the user's name and answers in a friendly manner.



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

Example #1

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

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

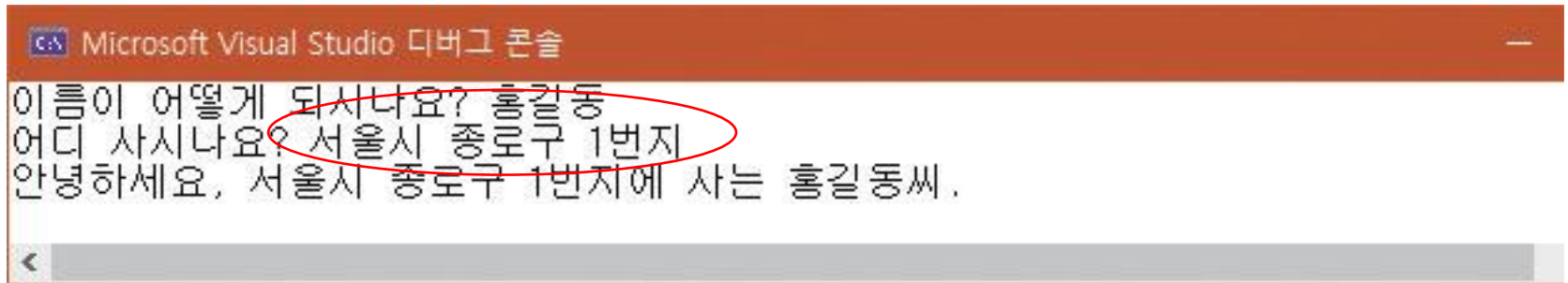
    printf("이름이 어떻게 되시나요? ");
    scanf("%s", name);

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

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

Example #2

- Let's write a program that receives the user's name and answers in a friendly manner.



Microsoft Visual Studio 디버그 콘솔

이름이 어떻게 되시나요? 홍길동
어디 사시나요? 서울시 종로구 1번지
안녕하세요, 서울시 종로구 1번지에 사는 홍길동씨.

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

문자 배열

최대 문자 개수

Example #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;
}
```

Interim check



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.



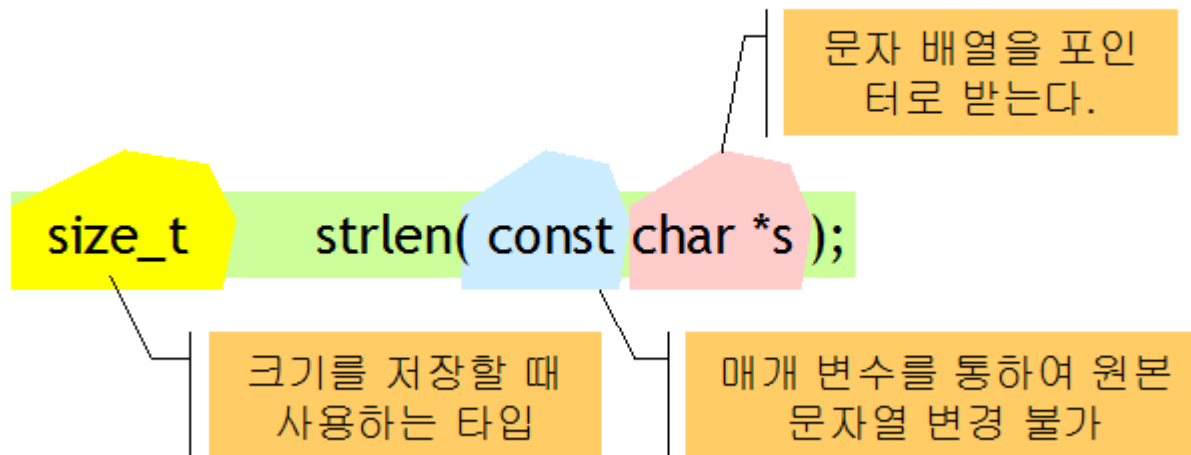
String processing library

- Using strings requires attaching two strings or comparing two strings to each other.
- String functions are declared in string.h. Therefore, to use these functions, string.h must be included at the beginning of the program.

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

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.

String length: strlen()



String length: 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

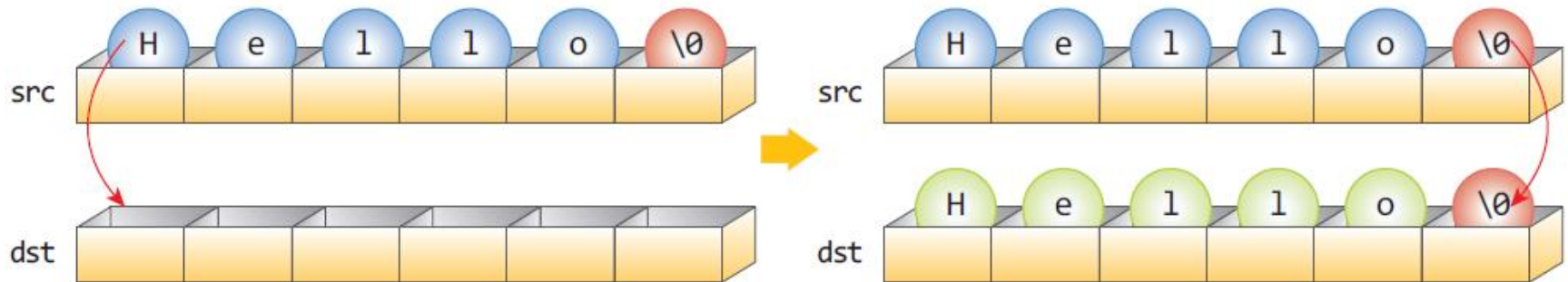
Copy String: strcpy()

```
#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;
}
```

Microsoft Visual Studio 디버그 콘솔

복사된 문자열 = Hello

Copy String: strcpy()

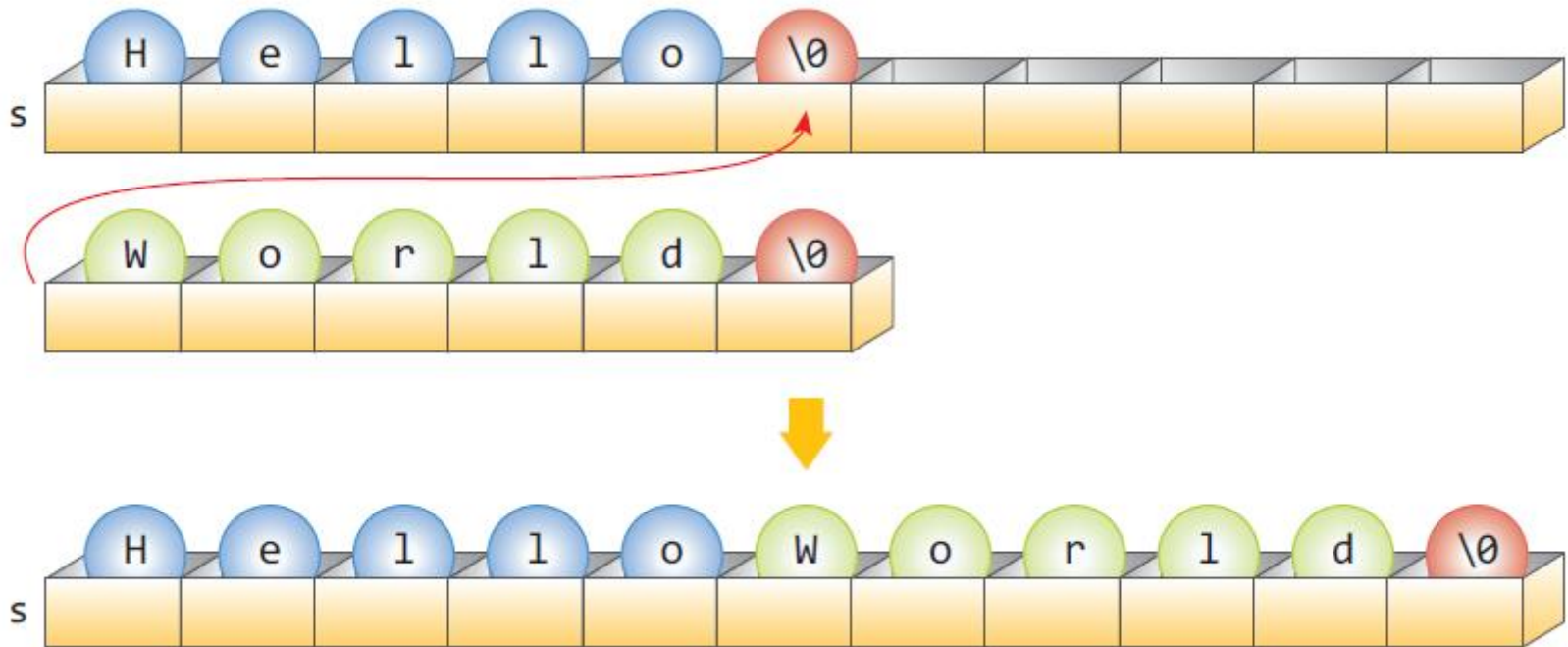


String Connections

```
#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;
}
```



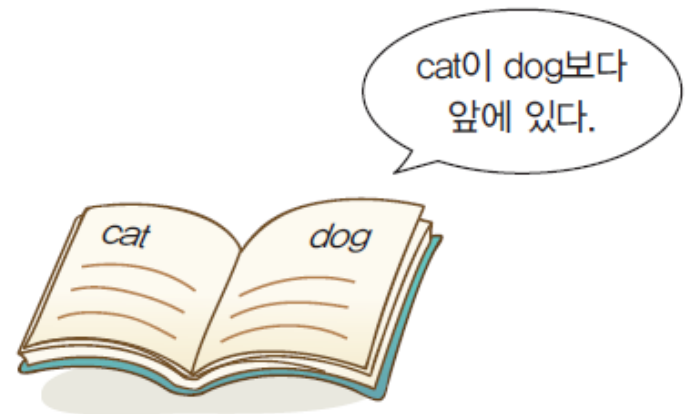
String Connections



String Comparison

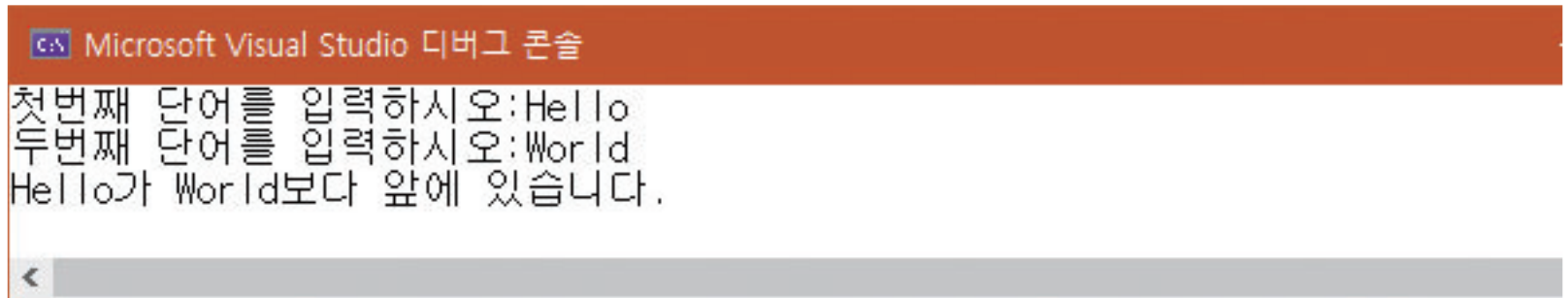
```
int strcmp( const char *s1, const char *s2 );
```

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



Example

- Let's write a program that receives two strings from the user and compares them.



```
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>

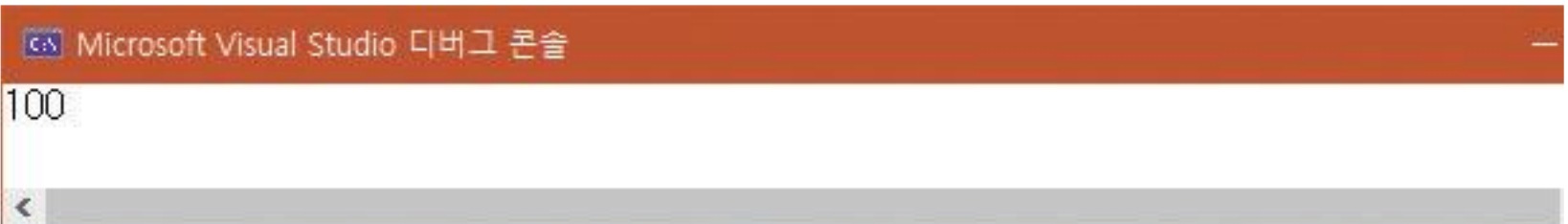
int main(void)
{
    char s1[80]; // 첫 번째 단어를 저장할 문자배열
    char s2[80]; // 두 번째 단어를 저장할 문자배열
    int result;

    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;
}
```

Converting a String to a Numeric

```
#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.
    return 0;
}
```



Check



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 )
```

Check



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.");
```

Interim check



중간점검

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

Lab: Short Answer Quiz Scoring

- Let's write a program to grade short answer questions. Repeat until the user enters the correct answer. Since it is a short answer question, check if the strings match

```
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] = "";

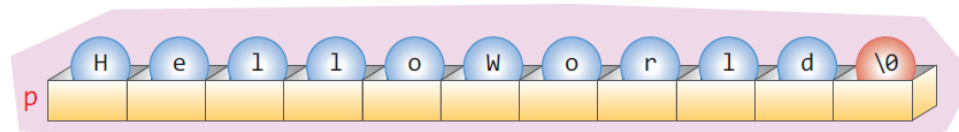
    do {
        printf("임베디드 장치에 가장 많이 사용되는 언어는? ");
        gets_s(buffer, sizeof(buffer));
    } while (strcmp(key, buffer) != 0);

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

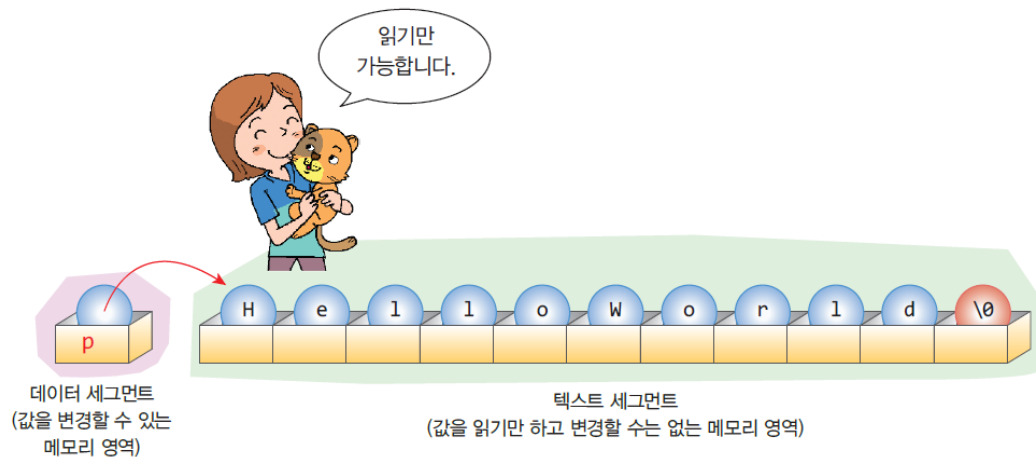
String and Pointer

- How are the two sentences below different?

```
char s[] = "HelloWorld"; // ①  
char *p = "HelloWorld"; // ②
```

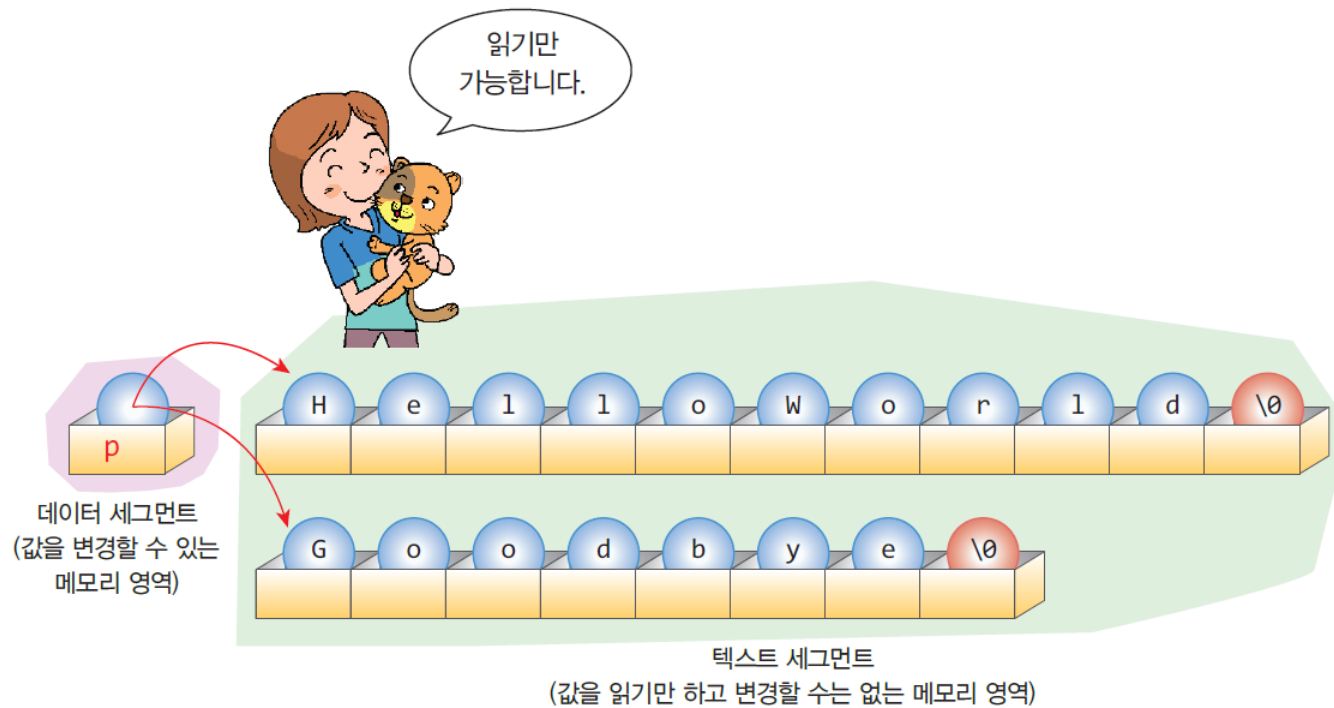


데이터 세그먼트
(값을 변경할 수 있는 메모리 영역)



String and Pointer

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



Example

```
#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;
}
```



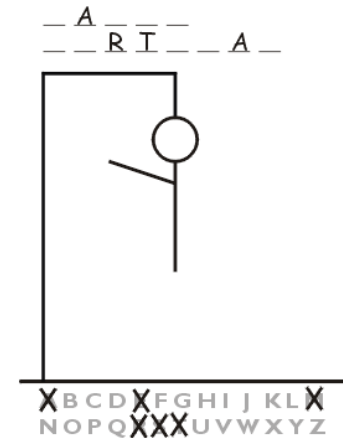
C:\> Microsoft Visual Studio 디버그 콘솔

포인터가 가리키는 문자열 = HelloWorld
포인터가 가리키는 문자열 = Goodbye

<

Lab: Hangman

- A game in which a character string composed of blanks is given and the user guesses the characters one by one

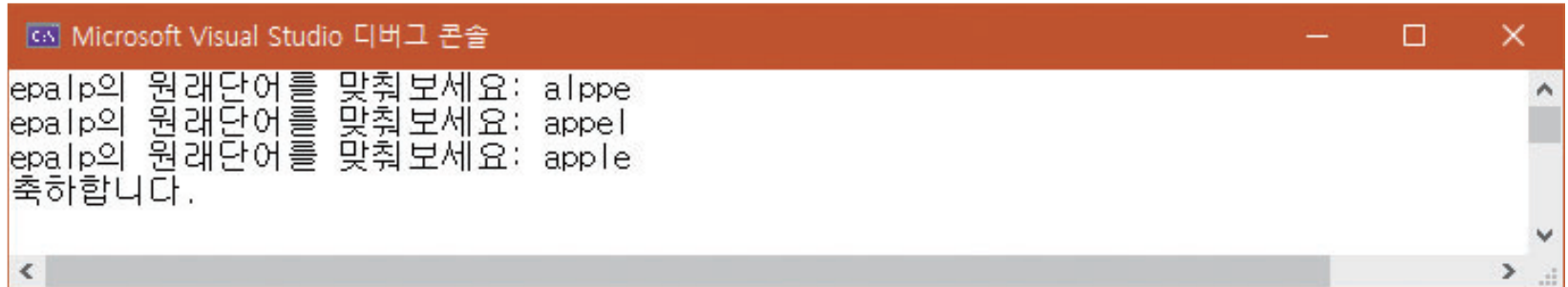


Sol:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(void) {
    char solution[100] = "meet at midnight";
    char answer[100] = "____ _";
    char ch;
    int i;
    while (1) {
        printf("\nAnswer: %s \n", answer);
        printf("Guess a character: ");
        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;
    }
    return 0;
}
```

Lab: Word Anagram Game

- Anagram is a game in which letters that make up English words are mixed up and rearranged in order.

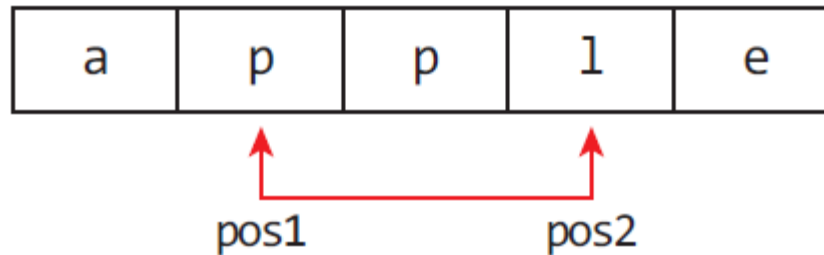


```
Microsoft Visual Studio 디버그 콘솔
epa|p의 원래단어를 맞춰 보세요: alppe
epa|p의 원래단어를 맞춰 보세요: appel
epa|p의 원래단어를 맞춰 보세요: apple
축하합니다.
```

The screenshot shows a Windows-style window titled "Microsoft Visual Studio 디버그 콘솔" (Microsoft Visual Studio Debug Console). The window has a red title bar with standard minimize, maximize, and close buttons. The content area is white and displays four lines of text in Korean. The first three lines are prompts for an anagram game: "epa|p의 원래단어를 맞춰 보세요:" followed by the scrambled words "alppe", "appel", and "apple". The fourth line is a congratulatory message: "축하합니다." (Congratulations). A vertical scrollbar is on the right side of the text area, and a horizontal scrollbar is at the bottom.

Lab: Word Anagram Game

- Random numbers are required to mix the characters in a string. Two random numbers can be generated and the letters at that location can be exchanged. Repeat this by the length of the string. Of course, the range of random numbers must be within the string.

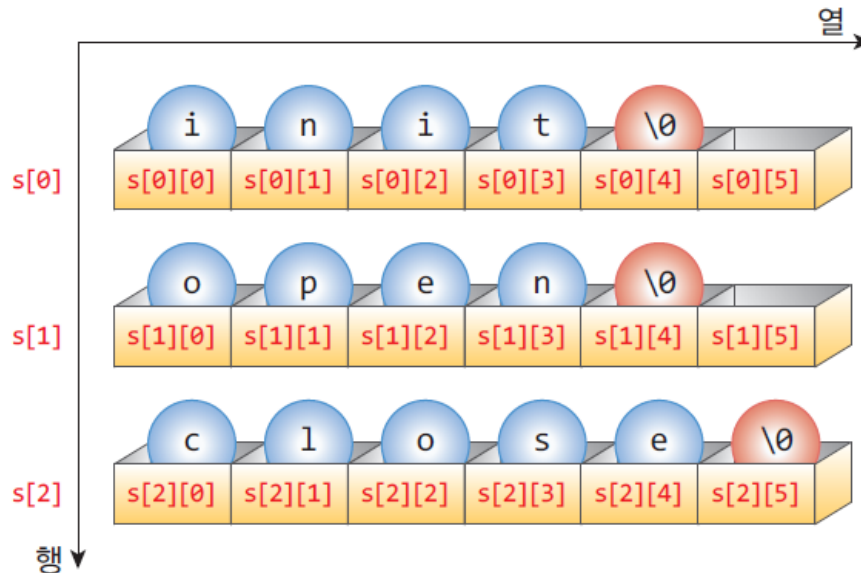


So

```
#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("Guess %s: ", s);
        scanf("%s", ans);
    } while (strcmp(ans, SOL) != 0);
    printf("Congratulation. \n");
    return 0;
}
```

Array of strings

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



Lab: the implementation of a Korean-English dictionary

- Let's implement a simple Korean-English dictionary using a three-dimensional character arrangement. English words are stored in `dic[i][0]`. The Korean description is stored in `dic[i][1]`.



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

The screenshot shows a Windows-style window titled "Microsoft Visual Studio 디버그 콘솔" (Microsoft Visual Studio Debug Console). The window has a red title bar with standard minimize, maximize, and close buttons. The main area is white and contains two lines of text: "단어를 입력하시오:rain" (Enter word: rain) and "rain: 비" (rain: rain). A horizontal scrollbar is visible at the bottom of the text area.

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: Sort String

- Can the strings be sorted alphabetically?

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;  
}
```