

21장 뜻자와 뜻자역 관련함수

차례

1. 스트림과 데이터의 이동
2. 문자단위 입출력 함수
3. 문자열 단위 입출력 함수
4. 표준 입출력 버퍼
5. 입출력이외의 문자열 관련 함수



C언어 표준라이브러리

3

- 라이브러리 : 프로그래밍언어에서 자주 사용하는 기능을 함수로 만들어 모아 놓은 것
- 라이브러리 종류
 - ▣ C언어 표준 라이브러리 : C언어에서 제공하는 함수들의 모임
 - 표준 입출력(`printf`, `scanf`, `getchar`, `putchar` 등) -> `stdio.h`
 - 수학 연산 (`sqrt`, `sin`, `cos` 등) -> `math.h`
 - 문자열 처리 (`strcpy`, `strlen` 등) -> `string.h`
 - 데이터 검색과 정렬(`qsort` 등) -> `stdlib.h`
 - 시간 처리(`time` 등) -> `time.h`
 - ▣ 사용자 정의 라이브러리 : 사용자가 직접 작성한 함수들의 모임

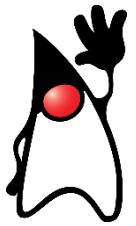


typedef

4

- 자료형 재정의(typedef) : 기존의 자료형(type)을 새로운 이름으로 재정의(define)해서 사용하는 것
- 비트수를 표시하거나 긴 자료형 이름을 한단어로 간단히 표현 가능
 - float -> FLOAT32, double -> FLOAT64, unsigned int -> UINT

```
typedef unsiged char    BYTE;
BYTE index;                // unsiged char index;와 같다.
typedef int    INT32;
INT32 i;                  // int i;와 같다.
typedef unsigned int    UINT32;
UINT32 k;                  // unsigned int k;와 같다.
```



문자열의 길이를 반환하는 함수: strlen

5

```
#include <string.h>
size_t strlen( const char *str );
```

- 전달된 문자열의 길이를 반환, 널문자는 길이에 포함하지 않음
- size_t -> typedef unsigned int size_t;
- string.h 안에 선언되어 있음

```
#include <string.h>
char str[ ] = "1234567";
printf("문자열 길이: %d\n", strlen(str));           // 7
printf("문자열 길이: %d\n", strlen("1234567"));    // 7
```

- strlen("1234567"); -> 문자열상수는 메모리에 저장되고 그것의 시작 주소로 치환됨



예제 1

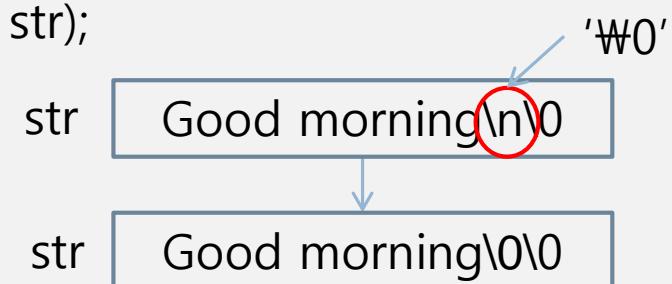
6

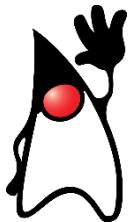
```
#include <stdio.h>
#include <string.h>
void RemoveBSN(char str[]);
int main(void)
{
    char str[100];
    printf("문자열 입력: ");
    fgets(str, sizeof(str), stdin);
    printf("길이: %d, 내용: %s \n", strlen(str), str);

    RemoveBSN(str);
    printf("길이: %d, 내용: %s \n", strlen(str), str);
    return 0;
}
void RemoveBSN(char str[])
{
    int len = strlen(str);
    str[len - 1] = '\0';
}
```

문자열 입력: Good morning <엔터>
길이: 13, 내용: Good morning

길이: 12, 내용: Good morning



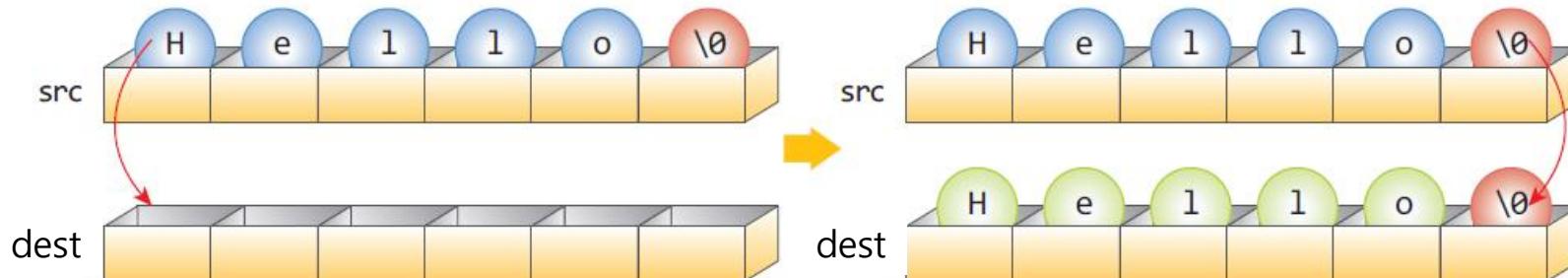


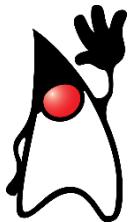
문자열을 복사하는 함수들: strcpy, strncpy

7

```
#include <string.h>
char *strcpy( char *dest, const char *src );
char *strncpy( char *dest, const char *src, size_t n );
```

- strcpy함수는 src에 저장된 문자열을 dest에 널문자 포함하여 복사 -> 배열 dest의 범위를 넘어서 복사할 위험이 있음
- 복사된 문자열(dest)의 주소값 반환
- strncpy함수는 src에 저장된 문자열을 dest에 최대 n개의 문자만 복사-> 널문자가 포함 안되면 별도로 복사해야 함





문자열을 복사하는 함수들: strcpy, strncpy

8

```
#include <string.h>
char str1[30] = "Simple String";
char str2[30];
strcpy(str2, str1);
str2 = "hello"; // error
```

- *str1*에서 널문자를 포함하여 *str2*로 복사함
- 목적지 *str2*의 크기가 작은 경우 실행오류 발생함
- 문자열 대입시는 *strcpy*함수 이용

```
#include <string.h>
char str1[30] = "Simple String";
char str2[30];
strncpy(str2, str1, sizeof(str2));
```

- *str1*에서 30개 문자를 *str2*로 복사,
- *str1*의 널문자가 30개 안에 포함되면 문제없음
- 널문자가 30개 안에 포함되지 않으면 복사안됨 -> 널문자 추가로 복사해야 함



예제 2: strncpy 함수를 잘못 사용한 예

9

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str1[20] = "1234567890";
    char str2[20];
    char str3[5];
    strcpy(str2, str1);
    puts(str2);
    strncpy(str3, str1, sizeof(str3));
    puts(str3);
    strncpy(str3, str1, sizeof(str3) - 1);
    str3[sizeof(str3) - 1] = '\0';
    puts(str3);
    return 0;
}
```

1234567890

12345_deposit_deposit_deposit_deposit_deposit_deposit_deposit_deposit
1234

- 배열 길이 str1에 딱 맞는 길이만큼만 복사를 하겠다는 의도의 문장
- 두 번째 strncpy 함수 호출 후의 결과는 널 문자가 복사되지 않았기 때문
- 문자열을 복사할 때에는 항상 널 문자의 복사까지 고려해야 함

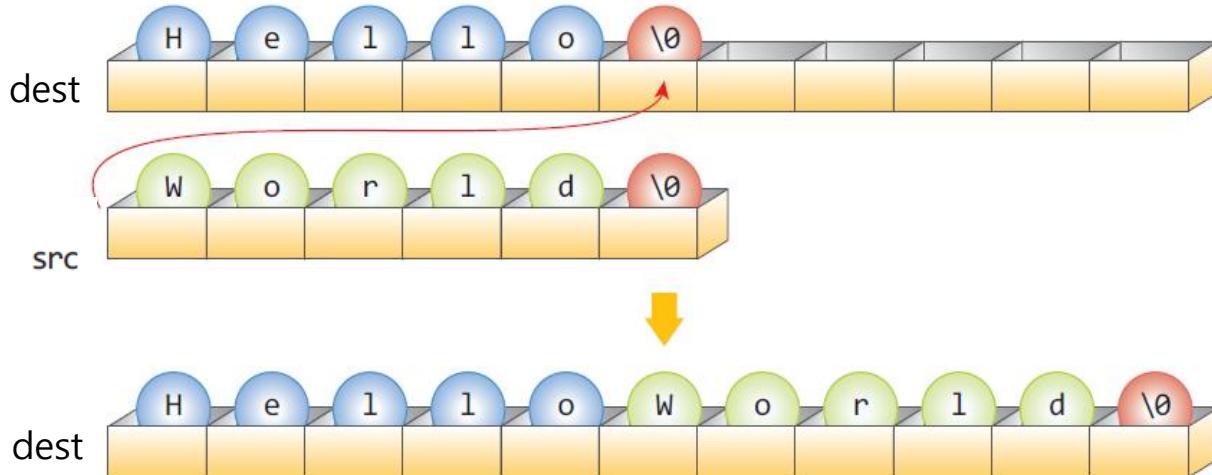


문자열을 더붙이는 함수들: strcat, strncat

10

```
#include <string.h>
char *strcat( char *dest, const char *src );
char *strncat( char *dest, const char *src, size_t n );
```

- strcat 함수는 문자열 src를 문자열 dest에 연결함
- strncat 함수는 연결할 문자열의 최대 길이를 제한함 -> 최대 n개의 문자를 연결하되 널 문자 포함하여 n+1개의 문자를 연결함





문자열을 덧붙이는 함수들: strcat, strncat

11

```
char str1[30] = "First~";  
char str2[30] = "Second";  
strcpy(str1, str2);
```

```
char str1[30] = "First~";  
char str2[30] = "Second";  
strncat(str1, str2, 8);
```

str1의 뒤에 str2를 복사 -> str1에 충분히 큰 공간이 없는 경우 실행 오류 발생

str1의 뒤에 str2의 문자열 중에 8개를 덧붙이고 널문자를 자동으로 추가해줌





예제 3

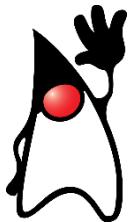
12

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str1[20] = "First~";
    char str2[20] = "Second";
    char str3[20] = "Simple num: ";
    char str4[20] = "1234567890";

    strcat(str1, str2);
    puts(str1);

    strncat(str3, str4, 7);
    puts(str3);
    return 0;
}
```

First~Second
Simple num: 1234567



문자열을 비교하는 함수들: strcmp, strncmp

13

```
#include <string.h>
int strcmp( const char *s1, const char *s2 );
int strncmp( const char *s1, const char *s2, size_t n );
```

- s1, s2의 문자열이 같으면 0을 반환, s1이 크면 0보다 큰 값 반환, s2 가 크면 0보다 작은 값 반환
- 문자열의 크기는 아스키코드 값을 기준으로 비교, 즉, 사전편찬순서 를 기준으로 뒤에 위치할수록 더 큰 문자열로 인식 -> "abc" > "ABC", "hello" < "world"
- strncmp는 앞에서 최대 n개의 문자만을 비교함

```
printf("%d\n", strcmp("ABCD", "ABCC"));           // 1 출력
printf("%d\n", strcmp("ABCD", "ABCDE"));           // -1 출력
```



예제 4

14

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str1[20];
    char str2[20];
    printf("문자열 입력 1: "); scanf("%s", str1);
    printf("문자열 입력 2: "); scanf("%s", str2);

    if (!strcmp(str1, str2))           // if(str1 == str2) -> 논리적 오류
        puts("두 문자열은 완벽히 동일합니다.");
    else
    {
        puts("두 문자열은 동일하지 않습니다.");
        if (!strncmp(str1, str2, 3)) puts("그러나 앞 세 글자는 동일합니다.");
    }
    return 0;
}
```

문자열 입력 1: simple
문자열 입력 2: simon
두 문자열은 동일하지 않습니다.
그러나 앞 세 글자는 동일합니다.

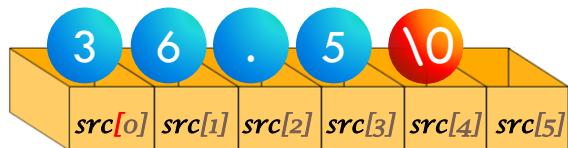


문자열을 수치로 변환

15

함수	설명
<code>int atoi(const char *str);</code>	<code>str</code> 을 <code>int</code> 형으로 변환한다.
<code>long atoi(const char *str);</code>	<code>str</code> 을 <code>long</code> 형으로 변환한다.
<code>double atof(const char *str);</code>	<code>str</code> 을 <code>double</code> 형으로 변환한다.

- 문자열에 저장된 숫자 정보를 `int`, `long`, `double`형 수치로 변환
- 헤더파일 `stdlib.h`에 선언



`char src[6] = "36.5"; -> 문자열`



`double v -> 실수`



예제 5

16

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    char str[20];
    printf("정수 입력: ");
    scanf("%s", str);
    printf("%d \n", atoi(str));

    printf("실수 입력: ");
    scanf("%s", str);
    printf("%g \n", atof(str));
    return 0;
}
```

정수 입력: 1523 <엔터>

1523

실수 입력: 12.45 <엔터>

12.45

char str[20]

'1'	'5'	'2'	'3'	'\0'
-----	-----	-----	-----	------	------

int a

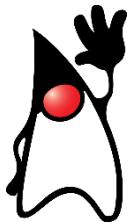
1523

char str[20]

'1'	'2'	'.'	'4'	'5'	'\0'
-----	-----	-----	-----	-----	------	------

double a

15.45

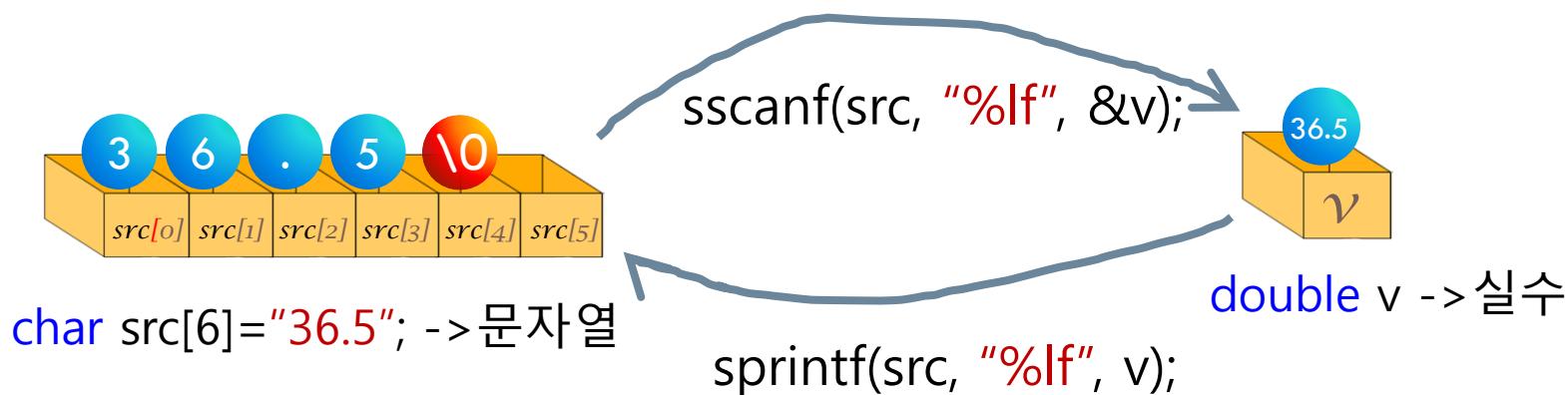


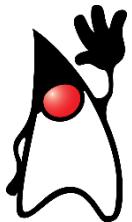
수치를 문자열로 변화

17

함수	설명
<code>int sscanf(const char *str, const char *format, ...)</code>	문자열을 수치로 변환
<code>int sprintf(char* str, const char* format, ...);</code>	수치를 문자열로 변환

- 헤더파일 stdio.h에 선언





예제 6

18

```
#include <stdio.h>
int main(void)
{
    char src[10] = "3.14159";
    double value;
    sscanf(src, "%lf", &value);
    printf("%lf\n", value);
    value = 2*value;
    sprintf(src, "%lf", value);
    printf("%s\n", src);
    return 0;
}
```

3.141590
6.283180



- 키보드로부터 문자열을 입력 받아 이전문자열에 연결하여 화면에 출력해주는 프로그램을 작성하시오. "quit"를 입력하면 종료한다.
- 힌트: 무한루프와 break문 이용하라. 문자열 비교는 strcmp 이용

문자열 입력 : hello<엔터>

입력된 문자열: hello

문자열 입력 : world<엔터>

입력된 문자열: hello world

문자열 입력 : C++<엔터>

입력된 문자열: hello world C++

문자열 입력 : quit<엔터>



실습과제 2

20

- 예제 1번과 동일하게 동작하도록 my_strlen 함수를 작성하라.
- 널문자를 만날 때까지 문자의 개수를 카운트한다.

```
#include <stdio.h>
#include <string.h>
//my_strlen 함수 선언
int main(void)
{
    char str[100];
    printf("문자열 입력: ");
    fgets(str, sizeof(str), stdin);
    printf("길이: %d, 내용: %s \n", my_strlen(str), str);
    RemoveBSN(str);
    printf("길이: %d, 내용: %s \n", my_strlen(str), str);
    return 0;
}
//my_strlen 함수 정의
```

문자열 입력: Good morning <엔터>
길이: 13, 내용: Good morning

길이: 12, 내용: Good morning



실습과제 3

21

- 아래 실행결과처럼 동작하도록 my_strcpy함수의 선언과 정의를 추가하시오. 힌트: while문을 사용하여 널문자를 만날 때까지 문자를 하나씩 복사한다. 문자열의 마지막에는 널문자를 추가 저장한다.

```
#include <stdio.h>
#include <string.h>
//my_strcpy 함수 선언
int main(void)
{
    char str1[20] = "1234567890";
    char str2[20];
    my_strcpy(str2, str1);
    puts(str2);
    return 0;
}
//my_strcpy 함수 선언
```

1234567890



실습과제 4

22

- 아래 실행결과처럼 동작하도록 my_strcat 함수의 선언과 정의를 추가 하시오. 힌트: while문을 사용하여 널문자를 만날 때까지 문자를 이어준다. 문자열의 마지막에는 널문자를 추가 저장한다.

```
#include <stdio.h>
#include <string.h>
//my_strcat 함수 선언
int main(void)
{
    char str1[20] = "First~";
    char str2[20] = "Second";
    my_strcat(str1, str2);
    puts(str1);
    return 0;
}
//my_strcat 함수 정의
```

First~Second

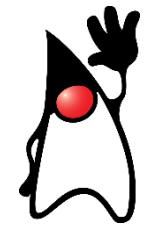


실습과제 5

23

- 단어 5개을 입력 받고 사전에서 제일 앞과 뒤에 나오는 단어를 각각 출력하시오.
- 문자열의 최대, 최소값은 각각 사전에서 제일 앞, 뒤에 나오는 단어임,
- 문자열의 대소는 strcmp함수 이용할 것

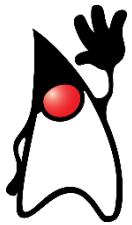
```
문자열 입력 : hello<엔터>
문자열 입력 : world<엔터>
문자열 입력 : apple<엔터>
문자열 입력 : melon<엔터>
문자열 입력 : banana<엔터>
제일앞 : apple
제일뒤 : world
```



실습과제 6

24

- 교재 445페이지 문제1,2,3중에서 1문제를 변형하여 푸시오. 그대로 풀면 점수 없음



21장 정리문제

25

- 1) 문자는 컴퓨터에서 어떻게 저장되는가?
- 2) 문자입력함수를 아는대로 설명하라.
- 3) 문자출력함수를 아는대로 설명하라.
- 4) 다음 문장을 실행할 때 문제점을 설명하라.

```
scanf("%d",&num);  
scanf("%c",&ch);
```

- 5) 4번의 해결방법을 설명하라. 게시판의 문자 및 문자열 입력시 엔터키 처리방법을 참고할 것
- 6) 널문자의 역할은?
- 7) 널문자를 자동으로 저장해주는 경우를 설명하라
- 8) scanf함수로 문자열을 입력받을 때 종료문자는?
- 9) gets함수로 문자열을 입력받을 때 종료문자는?
- 10) scanf와 gets함수의 차이점은?



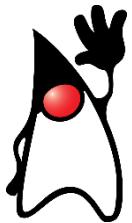
21장 정리문제

26

- 11) 문자열 출력함수를 설명하라.
- 12) C표준라이브러리 함수인 strcpy, strlen, strcat과 똑같은 기능을 갖는 함수를 작성하시오. 함수를 활용하는 예제까지 포함하여 만들것.
- 13) 문자열 저장시 아래 2문장 모두 가능합니다. 2가지방법의 차이를 설명하라.

```
char str[20] = "apple";
```

```
char *p = "pineapple";
```



과제제출방법

27

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡쳐할 때 글자를 알아보기 쉽게 확대해서 캡쳐할 것.
- 소스코드의 첫 부분은 아래처럼 제목, 날짜, 작성자(학번, 이름)를 작성할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2023년 9월 10일
// 작성자 : 15010101 홍길동
// *****

// 소스코드 작성
```