

-Part3-

제2장 문자열 표준 함수와 기타 표준 함수

학습목차

2.1 문자열 처리 함수 1

2.2 문자열 처리 함수 2

2.3 기타 표준 함수

2.1 문자열 처리 함수 1

2.1 문자열 처리 함수 1

▶ 배울 내용

- ① gets() 함수와 puts() 함수
- ② strlen() 함수 - 문자열의 길이를 알려준다
- ③ strcpy() 함수와 strncpy() 함수
- 문자열을 복사한다
- ④ strcat() 함수와 strncat() 함수
- 문자열을 결합한다
- ⑤ strcmp() 함수와 strncmp() 함수
- 문자열을 비교한다

2.1 문자열 처리 함수 1 (1/18)

▶ gets()함수와 puts()함수

- ✓ 헤더파일 : **stdio.h**
- ✓ **gets()**함수: 문자열을 입력받는 함수
- ✓ **puts()**함수: 문자열을 출력하는 함수

함수의 원형	예제	설명
<pre>#include<stdio.h> char* gets (char* s)</pre>	<pre>char array[10]; gets(array);</pre>	전달된 메모리 주소에 문자열 저장 성공 : 입력된 문자열 반환 실패 : NULL 포인터 반환
<pre>#include<stdio.h> int puts (const char* s)</pre>	<pre>char array[10] = "Good luck"; puts(array);</pre>	전달된 메모리 주소의 문자열 출력 성공 : 0값 또는 0이 아닌값 반환 실패 : EOF 반환

▶ EOF(End Of File)

- ✓ 파일의 끝을 의미
- ✓ stdio.h 에 -1로 매크로 상수 정의
- ✓ [ctr+z]를 입력하면 EOF로 인식

2.1 문자열 처리 함수 1 (2/15)

▶ gets()

✓ 문자열 입력

✓ [Enter]키(개행 문자 \n)를 입력 받으면...

- '문자열 끝에 자동으로 종료 문자(NULL 문자 \0)를 붙인다.'
- **주의** : 메모리 오버플로우가 발생하지 않게 문자열의 크기를 잘 할당 하자!
- 에러가 발생하면 NULL 포인터 반환

▶ puts()

✓ 문자열 출력

✓ 출력하면서 자동으로 [Enter]키(개행 문자 \n)를 삽입

✓ 에러가 발생하면 EOF(-1) 반환

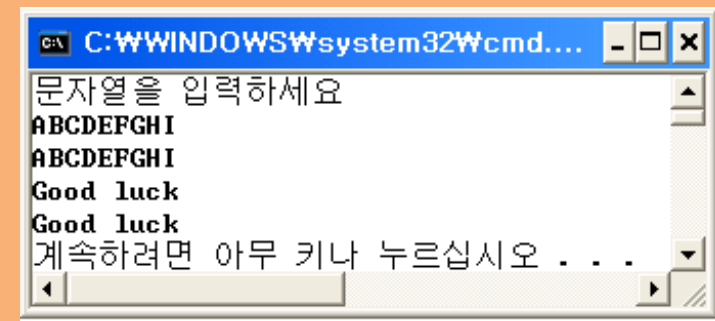
2.1 문자열 처리 함수 1 (3/15)---[2-1.c 실습]

```
#include <stdio.h>
int main(void)
{
    char array1[10];
    char array2[10] = "Good luck";

    puts("문자열을 입력하세요");
    gets(array1);          // 문자열 입력
    puts(array1);          // 문자열 출력

    puts(array2);          // 문자열 출력
    puts("Good luck");    // 문자열 출력

    return 0;
}
```



2.1 문자열 처리 함수 1 (4/15)---[2-2.c 실습]

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

```
int main(void)
```

```
{
```

```
    char array1[20];
```

```
    char array2[20];
```

```
    gets(array1);
```

```
// what is your name? 입력
```

```
    puts(array1);
```

```
// what is your name? 출력
```

```
    scanf ("%s",array2);
```

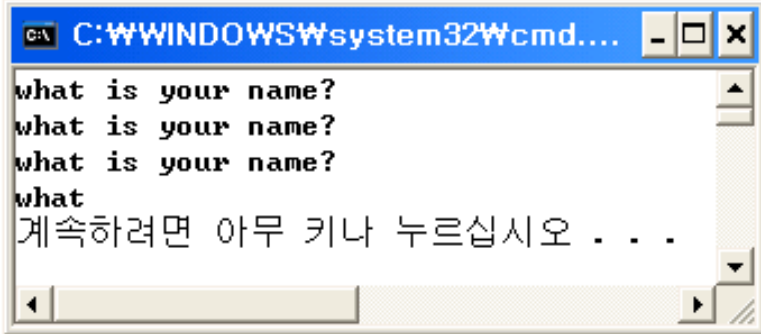
```
// what is your name? 입력
```

```
    printf("%s\n", array2);
```

```
// what 출력
```

```
    return 0;
```

```
}
```



```
C:\WINDOWS\system32\cmd...  
what is your name?  
what is your name?  
what is your name?  
what  
계속하려면 아무 키나 누르십시오 . . .
```


2.1 문자열 처리 함수 1

▶ 배울 내용

① gets() 함수와 puts() 함수

② strlen() 함수 - 문자열의 길이를 알려준다

③ strcpy() 함수와 strncpy() 함수
- 문자열을 복사한다

④ strcat() 함수와 strncat() 함수
- 문자열을 결합한다

⑤ strcmp() 함수와 strncmp() 함수
- 문자열을 비교한다

2.1 문자열 처리 함수 1 (5/15)

▶ strlen() 함수

- ✓ 헤더파일 : **string.h**
- ✓ 문자열의 길이를 알려주는 함수
- ✓ 주의 사항
 - '문자열의 끝을 알리는 종료문자('\0')는 길이에 포함되지 않는다.'

함수의 원형	예제	설명
<pre>#include<string.h> size_t strlen (const char* s)</pre>	<pre>char array[10] = "Good luck"; strlen(array);</pre>	전달된 메모리 주소 array 부터 종료문자를 만날때 까지 저장된 문자열의 길이를 반환

2.1 문자열 처리 함수 1 (6/15)---[2-3.c 실습]

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

int main(void)
{
    char array1[] = "Hello C";
    char array2[] = "안녕하세요";

    printf("영문 문자열의 길이: %d \\\n", strlen(array1) ); // 7 출력
    printf("한글 문자열의 길이  %d \\\n", strlen(array2) ); // 10 출력

    printf("영문 문자열의 길이: %d \\\n", sizeof(array1) ); // 8 출력
    printf("한글 문자열의 길이: %d \\\n", sizeof(array2) ); // 11 출력

    return 0;
}
```

2.1 문자열 처리 함수 1

▶ 배울 내용

- ① `gets()` 함수와 `puts()` 함수
- ② `strlen()` 함수 - 문자열의 길이를 알려준다
- ③ `strcpy()` 함수와 `strncpy()` 함수
- 문자열을 복사한다
- ④ `strcat()` 함수와 `strncat()` 함수
- 문자열을 결합한다
- ⑤ `strcmp()` 함수와 `strncmp()` 함수
- 문자열을 비교한다

2.1 문자열 처리 함수 1 (7/15)

▶ strcpy() 함수와 strncpy() 함수 - '문자열을 복사한다.'

✓ 헤더파일 : **string.h**

✓ **strcpy()** 함수

- 두 번째 인자의 문자열을 첫 번째 인자의 메모리 주소에 복사

✓ **strncpy()** 함수:

- 두 번째 인자의 문자열을 첫 번째 인자의 메모리 주소에 복사
- 세 번째 인자는 복사해야 할 크기

함수의 원형	예제	설명
<pre>#include<string.h> char* strcpy (char* dest, const char* src)</pre>	<pre>char array1[10] = "Good luck"; char array2[10]; strcpy(array2, array1);</pre>	array1의 문자열을 array2에 복사, 성공: 복사된 문자열의 시작 주소 반환
<pre>#include<string.h> char* strncpy (char* dest, const char* src, size_t n)</pre>	<pre>char array1[10] = "Good luck"; char array2[10]; strncpy(array2, array1, 3);</pre>	array1의 문자열을 array2에 3byte 만 복사, 성공: 복사된 문자열의 시작 주소 반환

2.1 문자열 처리 함수 1 (8/15)---[2-4.c 실습]

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

int main(void)
{
    char array1[12] = "Hello world";
    char array2[12];
    char array3[12];

    strcpy(array2, array1);
    strncpy(array3, array1, 12 );
    //strncpy(array3, array1, strlen(array1)+1 );

    puts(array2);
    puts(array3);

    return 0;
}
```

2.1 문자열 처리 함수 1 (9/15)---[2-5.c 실습]

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

int main(void)
{
    char array1[6] = "Hello";
    char array2[3];

    strncpy(array2, array1, 3);
    array2[2] = '\0';    // 종료 문자 삽입

    puts(array2);

    return 0;
}
```

2.1 문자열 처리 함수 1

▶ 배울 내용

- ① gets() 함수와 puts() 함수
- ② strlen() 함수 – 문자열의 길이를 알려준다
- ③ strcpy() 함수와 strncpy() 함수
– 문자열을 복사한다
- ④ strcat() 함수와 strncat() 함수
– 문자열을 결합한다
- ⑤ strcmp() 함수와 strncmp() 함수
– 문자열을 비교한다

2.1 문자열 처리 함수 1 (10/15)

▶ strcat() 함수와 strncat() 함수 - '문자열을 결합한다.'

✓ 헤더파일 : **string.h**

✓ **strcat()**

- 두 번째 인자의 문자열을 첫 번째 인자의 메모리 주소에 **결합**

✓ **strncat()**

- 두 번째 인자의 문자열을 첫 번째 인자의 메모리 주소에 **결합**
- 세 번째 인자는 결합해야 할 크기

함수의 원형	예제	설명
<pre>#include<string.h> char* strcat (char* dest, const char* src)</pre>	<pre>char array1[10] = "Good"; char array2 [5] = "luck"; strcat(array1, array2);</pre>	array1 에 array2 문자열을 결합 성공: 결합된 문자열의 시작 주소 반환
<pre>#include<string.h> char* strncat (char* dest, const char* src, size_t n)</pre>	<pre>char array1[10] = "Good"; char array2 [5] = "luck"; strncat(array1, array2, 3);</pre>	array1에 array2 문자열을 3byte 만큼 결합, 성공: 결합된 문자열의 시작 주소 반환

2.1 문자열 처리 함수 1 (11/15)---[2-6.c 실습]

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

```
int main(void)
{
    char array1[100];
    char array2[50];
```

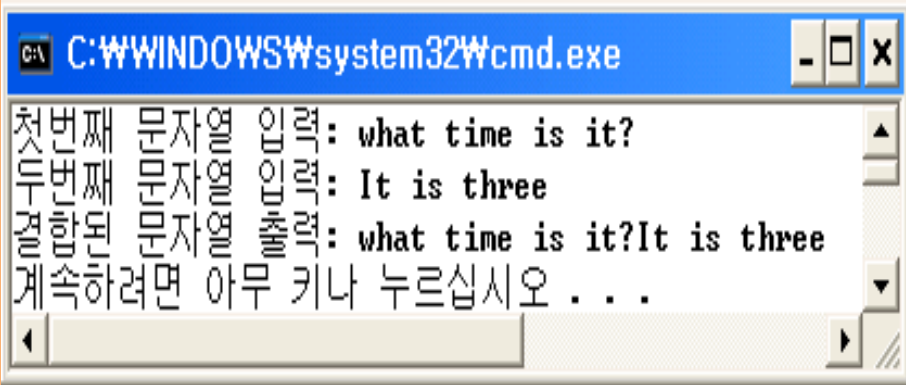
```
    printf("첫 번째 문자열 입력: ");
    gets(array1);
```

```
    printf("두 번째 문자열 입력: ");
    gets(array2);
```

```
    strcat(array1, array2); // 문자열 결합
```

```
    printf("결합된 문자열 출력: ");
    puts(array1);
    return 0;
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
첫번째 문자열 입력: what time is it?
두번째 문자열 입력: It is three
결합된 문자열 출력: what time is it?It is three
계속하려면 아무 키나 누르십시오 . . .
```

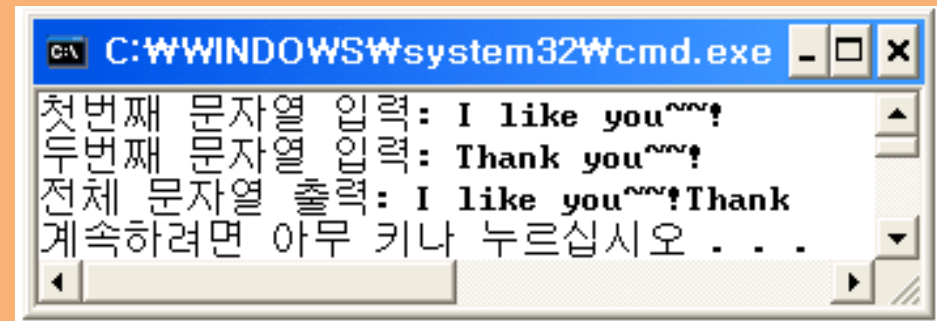
2.1 문자열 처리 함수 1 (12/15)---[2-7.c 실습]

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

```
int main(void)
{
    char array1[100];
    char array2[50];
```

```
    printf("첫 번째 문자열 입력: ");
    gets(array1);
```

```
    printf("두 번째 문자열 입력: ");
    gets(array2);
```



```
    strncat(array1, array2, 6); // 6 byte 만큼의 문자열 결합
```

```
    printf("전체 문자열 출력: ");
    puts(array1);
    return 0;
```

```
}
```

2.1 문자열 처리 함수 1

▶ 배울 내용

- ① `gets()` 함수와 `puts()` 함수
- ② `strlen()` 함수 - 문자열의 길이를 알려준다
- ③ `strcpy()` 함수와 `strncpy()` 함수
- 문자열을 복사한다
- ④ `strcat()` 함수와 `strncat()` 함수
- 문자열을 결합한다
- ⑤ `strcmp()` 함수와 `strncmp()` 함수
- 문자열을 비교한다

2.1 문자열 처리 함수 1 (13/15)

▶ strcmp() 함수와 strncmp() 함수 - '문자열을 비교한다.'

✓ 헤더파일 : **string.h**

✓ **strcmp()**

- 첫 번째 인자의 문자열과 두 번째 인자의 문자열을 비교

✓ **strncmp()**

- 첫 번째 인자의 문자열과 두 번째 인자의 문자열을 비교
- 세 번째 인자는 비교해야 할 크기

함수의 원형	예제	설명
<pre>#include<string.h> int strcmp (const char* s1, const char* s2)</pre>	<pre>char array1[10] = "Good"; char array2[10] = "luck"; strcmp(array1, array2);</pre>	array1의 문자열과 array2의 문자열을 비교합니다. 성공: array1 과 array2의 비교 결과를 반환
<pre>#include<string.h> int strncmp (const char* s1, const char* s2, size_t n)</pre>	<pre>char array1[10] = "Good"; char array2[10] = "luck"; strncmp(array1, array2, 3);</pre>	array1의 문자열과 array2의 문자열을 3개까지 비교합니다. 성공: array1 과 array2의 비교 결과를 반환

2.1 문자열 처리 함수 1 (14/15)

▶ strcmp()

- ✓ 첫 번째 인자의 문자열과 두 번째 인자의 문자열을 비교

▶ strncmp()

- ✓ 첫 번째 인자의 문자열과 두 번째 인자의 문자열을 비교
- ✓ 세 번째 인자는 비교해야 할 크기

▶ 비교 결과

반환값	설명
양수(0보다 큰 값)	array1의 문자열이 array2의 문자열보다 크다.
0	array1의 문자열이 array2의 문자열과 같다.
음수(0보다 작은 값)	array1의 문자열이 array2의 문자열보다 작다.

2.1 문자열 처리 함수 1 (15/15)---[2-8.c 실습]

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

int main(void)
{
    char array1[20] = "Good-morning";
    char array2[20] = "Good-afternoon";
    char array3[20] = "Good-evening";
    int result1, result2, result3;

    result1 = strcmp(array1, array2);    // 1 출력
    result2 = strncmp(array1, array2, 5); // 0 출력
    result3 = strcmp(array2, array3);    // -1 출력

    printf("%d %d %d \n", result1, result2, result3);
    return 0;
}
```

2.2 문자열 처리 함수 2

2.2 문자열 처리 함수 2

▶ 배울 내용

- ① `strchr()` 함수와 `strstr()` 함수
 - 문자와 문자열의 위치를 찾는다
- ② `strupr()` 함수와 `strlwr()` 함수
 - 문자열을 대문자로 바꾸거나 소문자로 바꾼다
- ③ `sscanf()` 함수와 `sprintf()` 함수
 - 메모리로부터 문자열을 입력받고 출력한다

2.2 문자열 처리 함수 2 (1/9)

▶ strchr() 함수와 strstr() 함수

- ✓ 헤더파일 : **string.h**
- ✓ **strchr()** 함수: **문자**의 위치를 찾는 함수
- ✓ **strstr()** 함수: **문자열**의 위치를 찾는 함수

함수의 원형	예제	성공
<pre>#include<string.h> char* strchr (const char* s, int c)</pre>	<pre>char array1[10] = "Good"; strchr(array1, 'd');</pre>	array1에서 문자 'd'의 메모리 주소를 찾는다. 성공: 찾은 문자 의 메모리 주소를 반환
<pre>#include<string.h> char* strstr (const char* s1, const char* s2)</pre>	<pre>char array1[10] = "Good-morning"; char array2[10] = "morning"; strstr (array1, array2);</pre>	array1에서 array2에 저장된 문자열을 찾는다. 성공: 찾은 문자열 의 메모리 주소를 반환

2.2 문자열 처리 함수 2 (2/9)---[2-9.c 실습]

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

int main(void)
{
    char array1[50] = "Good-morning, Good-afternoon, Good-evening";
    char array2[10] = "morning";
    char* p1=NULL;
    char* p2=NULL;

    p1 = strchr(array1, 'a');
    p2 = strstr(array1, array2);

    printf("문자 a의 위치 : %x \n", p1);
    printf("문자열 : %s \n", p1);
    printf("-----\n");
    printf("문자열의 시작 문자 위치: %x \n", p2);
    printf("문자열 : %s \n", p2);
    return 0;
}
```

2.2 문자열 처리 함수 2

▶ 배울 내용

- ① strchr() 함수와 strstr() 함수
 - 문자와 문자열의 위치를 찾는다
- ②strupr() 함수와 strlwr() 함수
 - 문자열을 대문자로 바꾸거나 소문자로 바꾼다
- ③ sscanf() 함수와 sprintf() 함수
 - 메모리로부터 문자열을 입력받고 출력한다

2.2 문자열 처리 함수 2 (3/9)

▶ **strupr()** 함수와 **strlwr()** 함수

- ✓ 헤더파일 : **string.h**
- ✓ **strupr()** 함수: 문자열을 **대문자**로 바꾸는 함수
- ✓ **strlwr()** 함수: 문자열을 **소문자**로 바꾸는 함수

함수의 원형	예제	설명
<code>#include<string.h></code> <code>char* strupr (const char* s)</code>	<code>char array1[10] = "good";</code> <code>strupr(array1);</code>	array1에서 저장된 문자열을 대문자로 변환 성공: 변환된 문자열의 시작 주소를 반환
<code>#include<string.h></code> <code>char* strlwr (const char* s)</code>	<code>char array1[10] = "GOOD";</code> <code>strlwr(array1);</code>	array1에서 저장된 문자열을 소문자로 변환 성공: 변환된 문자열의 시작 주소를 반환

2.2 문자열 처리 함수 2 (4/9)---[2-10.c 실습]

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

int main(void)
{
    char array1[50] = "good-morning,good-afternoon,good-evening";
    char array2[50] = "GOOD-MORNING,GOOD-AFTERNOON,GOOD-EVENING";
    char* p1=NULL;
    char* p2=NULL;

    p1 =strupr(array1);
    p2 =strlwr(array2);

    puts(p1);
    puts(p2);
    return 0;
}
```

2.2 문자열 처리 함수 2 (5/9)

▶ 참고

✓ **ctype.h**에 있는 문자 분류 함수

함수의 원형	설명
int isalnum (int c);	알파벳인지 숫자인지를 검사
int isalpha (int c);	알파벳인지 아닌지를 검사
int isdigit (int c);	숫자인지 아닌지를 검사
int islower (int c);	소문자인지 아닌지를 검사
int isupper (int c);	대문자인지 아닌지를 검사
int isspace (int c);	공백 문자인지 아닌지를 검사
int isxdigit (int c);	16진수인지 아닌지를 검사

2.2 문자열 처리 함수 2 (6/9)

▶ **ctype.h**에 있는 문자 분류 함수

```
#include <stdio.h>
#include <ctype.h>
int main(void)
{
    char a1='A', a2='a', a3='B';
    int result1, result2, result3;

    result1=isalnum(a1);
    result2=isupper(a2);
    result3=islower(a3);

    printf("%d %d %d \n", result1, result2, result3);

    return 0;
}
```


2.2 문자열 처리 함수 2

▶ 배울 내용

- ① `strchr()` 함수와 `strstr()` 함수
 - 문자와 문자열의 위치를 찾는다
- ② `strupr()` 함수와 `strlwr()` 함수
 - 문자열을 대문자로 바꾸거나 소문자로 바꾼다
- ③ `sscanf()` 함수와 `sprintf()` 함수
 - 메모리로부터 문자열을 입력받고 출력한다

2.2 문자열 처리 함수 2 (7/9)

▶ sscanf() 함수와 sprintf() 함수

- ✓ 헤더파일 : **string.h**
- ✓ **sscanf()** 함수: 배열(메모리)로부터 문자열을 입력받는 함수
- ✓ **sprintf()** 함수: 배열(메모리)에 문자열을 출력하는 함수

함수의 원형	설명
<code>#include<string.h></code> <code>int sscanf (const char * s1, const char * s2, ...);</code>	메모리에서 데이터를 입력 받는다. 성공: 데이터의 개수 반환
<code>#include<string.h></code> <code>int sprintf (char * s1, const char * s2, ...);</code>	메모리에 데이터를 출력한다. 성공: 문자열의 길이 반환

2.2 문자열 처리 함수 2 (8/9)---[2-10.c 실습]

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

int main(void)
{
    char array[50] = "100 3.14 good-morning";
    int num1;
    double num2;
    char str[50];

    //scanf("%d %lf %s",&num1, &num2, str);      // 키보드로부터 입력 받음
    sscanf(array,"%d %lf %s",&num1, &num2, str); // 배열로부터 입력 받음

    puts("출력");
    printf("%d, %lf, %s \n", num1, num2, str);
    return 0;
}
```

2.2 문자열 처리 함수 2 (9/9)---[2-12.c 실습]

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

int main(void)
{
    char array[50];
    int num1 = 100;
    double num2 = 3.14;
    char str[50] = "good-morning";

    printf("%d, %lf, %s \n", num1, num2, str);           // 모니터에 출력
    sprintf(array, "%d, %lf, %s \n", num1, num2, str); // 배열에 출력
    printf("%s \n", array);                             // 모니터에 출력

    return 0;
}
```

2.3 기타 표준 함수

2.3 기타 표준 함수

▶ 배울 내용

① 데이터 변환 표준 함수

② 수학 관련 표준 함수

2.3 기타 표준 함수 (1/11)

▶ 데이터를 변환하는 함수

✓ 헤더파일 : **stdlib.h**

함수의 원형	설명
double atof (const char* str);	문자열을 double형 데이터로 변환
int atoi (const char* str);	문자열을 int형 데이터로 변환
long atol (const char* str);	문자열을 long형 데이터로 변환

2.3 기타 표준 함수 (2/11)---[2-13.c 실습]

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    char* str1 = "3.14";
    char* str2 = "100";
    char* str3 = "100000000";

    double num1;
    int    num2;
    long   num3;

    num1 = atof(str1);
    num2 = atoi(str2);
    num3 = atol(str3);

    printf("%lf, %d, %ld \n", num1, num2, num3);
    return 0;
}
```


2.3 기타 표준 함수 (3/11)

▶ 데이터를 변환하는 함수

▶ 헤더파일 : **ctype.h**

함수의 원형	설명
int toascii (int num);	문자를 ASCII 문자 로 변환
int tolower (int num);	문자를 소문자 로 변환
int toupper (int num);	문자를 대문자 로 변환

2.3 기타 표준 함수 (4/11)---[2-14.c 실습]

```
#include <stdio.h>
#include <ctype.h>
int main(void)
{
    char a1='A';
    char a2='a';

    printf("아스키 코드 : %d \\\n", toascii(a1) ); // ASCII 값 65 출력
    printf("    소문자 : %c \\\n", tolower(a2) ); // 소문자 a 출력
    printf("    대문자: %c \\\n", toupper(a2) ); // 대문자 A 출력

    return 0;
}
```

2.3 기타 표준 함수

▶ 배울 내용

① 데이터 변환 표준 함수

② 수학 관련 표준 함수

2.3 기타 표준 함수 (5/11)

▶ 수학 관련 함수

▶ 헤더파일 : **math.h**

함수의 원형	설명
double ceil (double x);	X보다 큰 정수 반환
double floor (double x);	X보다 작은 정수 반환
double fabs (double x);	X의 절댓값 반환
double pow (double x, double y);	x^y
double sqrt (double x);	\sqrt{x}
double exp (double x);	e^x e는 자연 상수 (오일러의 수)
double log (double x);	$\log_e x$
double log10 (double x);	$\log_{10} x$

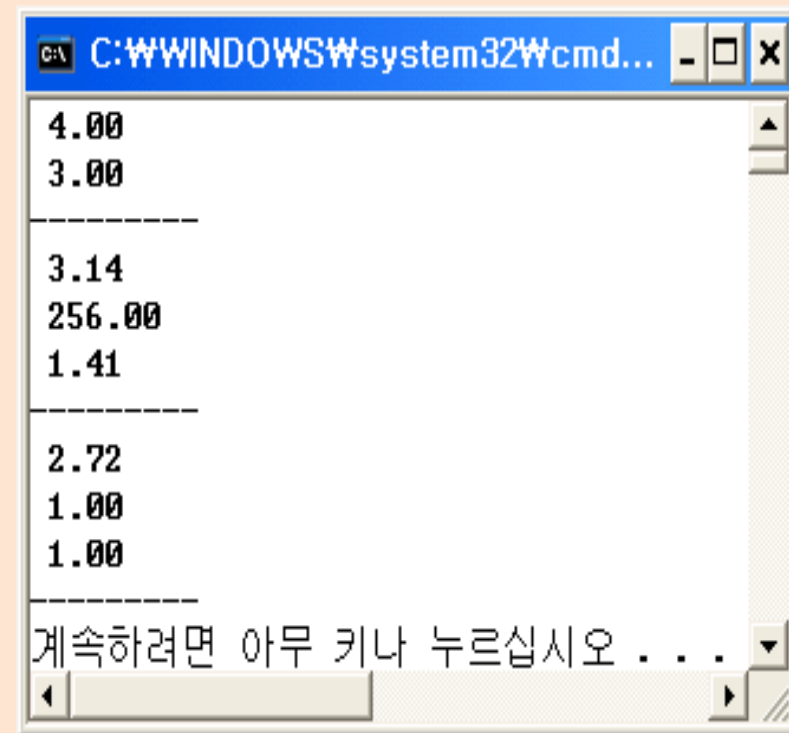
2.3 기타 표준 함수 (6/11)---[2-15.c 실습]

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    double a1 = 3.14;
    double a2 = -3.14;

    printf(" %.2lf \Wn", ceil(a1) );
    printf(" %.2lf \Wn", floor(a1) );
    printf("-----\Wn");

    printf(" %.2lf \Wn", fabs(a2) );
    printf(" %.2lf \Wn", pow(2,8) );
    printf(" %.2lf \Wn", sqrt(2) );
    printf("-----\Wn");

    printf(" %.2lf \Wn", exp(1) );
    printf(" %.2lf \Wn", log(exp(1)) );
    printf(" %.2lf \Wn", log10(10) );
    printf("-----\Wn");
    return 0;
}
```



```
C:\WINDOWS\system32\cmd...
4.00
3.00
-----
3.14
256.00
1.41
-----
2.72
1.00
1.00
-----
계속하려면 아무 키나 누르십시오 . . .
```

2.3 기타 표준 함수 (7/11)

▶ **math.h**에 있는 여러 삼각함수

함수의 원형	설명
double sin (double x);	삼각함수에서 x의 sin 값
double cos (double x);	삼각함수에서 x의 cos 값
double tan (double x);	삼각함수에서 x의 tan 값
double sinh (double x);	삼각함수에서 x의 sinh 값
double cosh (double x);	삼각함수에서 x의 cosh 값
double tanh (double x);	삼각함수에서 x의 tanh 값
double asin (double x);	삼각함수에서 x의 asin 값
double acos (double x);	삼각함수에서 x의 acos 값
double atan (double x);	삼각함수에서 x의 atan 값

2.3 기타 표준 함수 (8/11)

▶ rand() 함수와 srand() 함수 - '난수를 생성 시킨다.'

✓ 헤더파일 : **stdlib.h**

✓ rand() 함수: 난수를 생성시키는 함수

- '한가지 패턴으로 난수를 생성 시킨다.'

✓ srand() 함수: 난수의 패턴을 생성시키는 함수

- '여러 가지 패턴으로 난수를 생성 시킨다.'

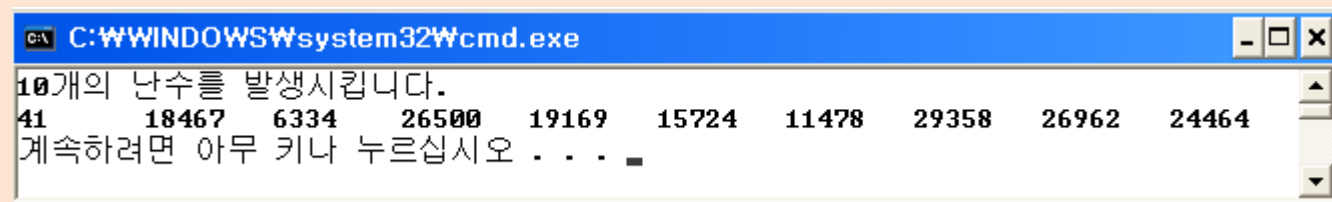
함수의 원형	설명
int rand (void);	난수를 생성
int srand (unsigned int seed);	seed 를 지정하여 난수를 생성 (seed: 난수 생성 패턴)

2.3 기타 표준 함수 (9/11)---[2-16.c 실습]

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i=0;
    puts("10개의 난수를 발생시킵니다.");

    while(i<10)
    {
        printf("%d \t", rand() );
        i=i+1;
    }

    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
10개의 난수를 발생시킵니다.
41      18467    6334    26500    19169    15724    11478    29358    26962    24464
계속하려면 아무 키나 누르십시오 . . .
```


2.3 기타 표준 함수 (10/11)---[2-17.c 실습]

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

int main(void)
{
    int i=0;
    puts("10개의 난수를 발생시킵니다.");
    srand(1); // 정수 값을 1~3까지 변경해 보자(패턴 발생)

    while(i<10)
    {
        printf("%d \t", rand());
        i=i+1;
    }
    return 0;
}
```

2.3 기타 표준 함수 (11/11)---[2-17.c 분석]

▶ srand(1)일 때

실행 결과 >>

41 18467 6334 26500 19169 15724 11478 29358 26962 24464

▶ srand(2)일 때

실행 결과 >>

45 29216 24198 17795 20494 10650 14590 26431 10705 18316

▶ srand(3)일 때

실행 결과 >>

48 7196 9294 9091 7031 23577 17702 23503 27217 12168

▶ rand()함수를 호출하면 기본적으로 srand(1) 설정

공부한 내용 떠올리기

▶ 다양한 문자열 처리 함수들

▶ 기타 표준 함수

- ✓ 데이터 변환 함수
- ✓ 수학함수

삶의 여유(출처: 사랑과 지혜의 탈무드)

