

CHAPTER 4

문자 처리: 문자 분류 및
변환 실습

실습 1

문제 1)

문자열 복사 함수 만들기 StrCpy

```
void main( void )
{
    char string[100];

    strcpy( string, "C 언어 길라잡이" );
    puts( string );

    StrCpy( string, "정보문화사" );
    puts( string );
}
```

```
C 언어 길라잡이
정보문화사
```

실습 1 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrCpy( char* dest, const char* src );
5
6  void main( void )
7  {
8      char string[100];
9
10     strcpy( string, "c 언어 길라잡이" );
11     puts( string );
12
13     StrCpy( string, "정보문화사" );
14     puts( string );
15 }
16
17 char* StrCpy( char* dest, const char* src )
18 {
19     char* pdest = dest;
20
21     while( *src )
22     {
23         *dest++ = *src++;
24     }
25
26     *dest = '\0';
27
28     return pdest;
29 }
30
```

실습 2

문제 2)

문자열 비교 함수 만들기 StrCmp

```
void main( void )
{
    int result;

    result = strcmp( "abc", "ABC" );
    printf( "%d \n", result );

    result = StrCmp( "abc", "abcde" );
    printf( "%d \n", result );

    result = StrCmp( "World Cup", "world cup" );
    printf( "%d \n", result );
}
```

```
1
-1
-1
```

실습 2 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  int StrCmp( const char* str1, const char* str2 );
4  void main( void )
5  {
6      int result;
7
8      result = strcmp( "abc", "ABC" );
9      printf( "%d \n", result );
10
11     result = StrCmp( "abc", "abcde" );
12     printf( "%d \n", result );
13
14     result = StrCmp( "World Cup", "world cup" );
15     printf( "%d \n", result );
16 }
17 int StrCmp( const char* str1, const char* str2 )
18 {
19     while( *str1 && *str2 )
20     {
21         if( *str1 != *str2 )
22         {
23             return ( *str1 < *str2 ) ? -1 : 1;
24         }
25         str1++, str2++;
26     }
27     if( *str1 != *str2 )
28     {
29         return ( *str1 < *str2 ) ? -1 : 1;
30     }
31     return 0;
32 }
```



```
1
-1
-1
```

실습 3

문제 3)

문자열 연결 함수 만들기 StrCat

```
void main( void )
{
    char string[100];

    strcpy( string, "나는 " );
    strcat( string, "프로그래머입니다." );
    puts( string );

    strcpy( string, "나는 " );
    StrCat( string, "프로그래머입니다." );
    puts( string );
}
```

나는 프로그래머입니다.

나는 프로그래머입니다.

실습 3 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrCat( char* dest, const char* src );
5
6  void main( void )
7  {
8      char string[100];
9
10     strcpy( string, "나는 " );
11     strcat( string, "프로그래머입니다." );
12     puts( string );
13
14     strcpy( string, "나는 " );
15     StrCat( string, "프로그래머입니다." );
16     puts( string );
17 }
18 char* StrCat( char* dest, const char* src )
19 {
20     char* pdest = dest;
21
22     dest += strlen(dest);
23
24     while( *src )
25     {
26         *dest++ = *src++;
27     }
28
29     *dest = '\0';
30
31     return pdest;
32 }
```

나는 프로그래머입니다.
나는 프로그래머입니다.

실습 4

문제 4)

문자열 검사 함수 만들기 StrChr

```
void main( void )
{
    char string[100];
    char* pos;

    strcpy( string, "This is a book." );
    pos = strchr( string, 'a' );
    printf( "%d 위치에 a가 있음 \n", pos - string );

    strcpy( string, "This is a book." );
    pos = StrChr( string, 'a' );
    printf( "%d 위치에 a가 있음 \n", pos - string );
}
```

```
8 위치에 a가 있음
8 위치에 a가 있음
```


실습 4 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrChr( const char* pstring, char ch );
5
6  void main( void )
7  {
8      char string[100];
9      char* pos;
10
11     strcpy( string, "This is a book." );
12     pos = strchr( string, 'a' );
13     printf( "%d 위치에 a가 있음 \n", pos - string );
14
15     strcpy( string, "This is a book." );
16     pos = StrChr( string, 'a' );
17     printf( "%d 위치에 a가 있음 \n", pos - string );
18 }
19 char* StrChr( const char* pstring, char ch )
20 {
21     while( *pstring && *pstring != ch )
22     {
23         pstring++;
24     }
25     if( *pstring == ch )
26     {
27         return (char*)pstring;
28     }
29     return (char*)0;
30 }
```

8 위치에 a가 있음
8 위치에 a가 있음

실습 5

문제 5)

문자열 뒤집는 함수 만들기 StrRev

```
void main( void )
{
    char string[100];
    char* pos;

    strcpy( string, "abcde 12345" );
    pos = StrRev( string );
    puts( string );
}
```

54321 edcba

실습 5 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrRev( char* pstring );
5  void main( void )
6  {
7      char string[100];
8      char* pos;
9
10     strcpy( string, "abcde 12345" );
11     pos = StrRev( string );
12     puts( string );
13 }
14 char* StrRev( char* pstring )
15 {
16     char *start = pstring;
17     char *left  = pstring;
18     char ch;
19
20     while( *pstring++ );
21
22     pstring -= 2;
23
24     while( left < pstring )
25     {
26         ch      = *left;
27         *left++ = *pstring;
28         *pstring-- = ch;
29     }
30     return start;
31 }
```

54321 edcba

실습 6

문제 6)

문자열 대문자로 변환하는 함수 만들기 StrUpr

```
void main( void )
{
    char string[100];

    strcpy( string, "beautiful girl." );
    StrUpr( string );
    puts( string );
}
```

BEAUTIFUL GIRL.

실습 6 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrUpr( char* pstring );
5
6  void main( void )
7  {
8      char string[100];
9
10     strcpy( string, "beautiful girl." );
11     StrUpr( string );
12     puts( string );
13 }
14 char* StrUpr( char* pstring )
15 {
16     char *cp;
17
18     for( cp = pstring; *cp; ++cp )
19     {
20         if( 'a' <= *cp && *cp <= 'z' )
21         {
22             *cp += 'A' - 'a';
23         }
24     }
25     return pstring;
26 }
```

BEAUTIFUL GIRL.

실습 7

문제 7)

문자열 왼쪽 공백 제거 함수 만들기 StrLTrim

isspace() 함수를 사용해 공백 체크

```
void main( void )
{
    char string[100];
    char* pstr;

    strcpy( string, "\n\t abcde" );
    pstr = StrLTrim( string );
    puts( pstr );
}
```

abcde

실습 7 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4
5  char* StrLTrim( char* pstring );
6
7  void main( void )
8  {
9      char string[100];
10     char* pstr;
11
12     strcpy( string, "\n\t abcde" );
13     pstr = StrLTrim( string );
14     puts( pstr );
15 }
16
17 char* StrLTrim( char* pstring )
18 {
19     while( *pstring )
20     {
21         if( !isspace(*pstring) )
22         {
23             return pstring;
24         }
25         pstring++;
26     }
27     return (char*)"";
28 }
```

abcde

실습 8

문제 8)

문자열 오른쪽 공백 제거 함수 만들기 StrRTrim

isspace() 함수를 사용해 공백 체크

```
void main( void )
{
    char string[100];
    char* pstr;

    strcpy( string, "abcde \t\n " );
    pstr = StrRTrim( string );
    printf( "[%s] \n", pstr );
}
```

abcde

실습 8 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4
5  char* StrRTrim( char* pstring );
6
7  void main( void )
8  {
9      char string[100];
10     char* pstr;
11
12     strcpy( string, "abcde \t\n " );
13     pstr = StrRTrim( string );
14     printf( "[%s] \n", pstr );
15 }
16 char* StrRTrim( char* pstring )
17 {
18     char* start = pstring;
19     while( *pstring )
20     {
21         pstring++;
22     }
23     pstring--;
24
25     while( start <= pstring )
26     {
27         if( !isspace(*pstring) )
28         {
29             *++pstring = '\0';
30             return start;
31         }
32         pstring--;
33     }
34     return (char*)" ";
35 }
```

abcde

실습 9

문제 9)

키보드로 입력한 문장에서 원하는 문자열을 찾아 그 개수를 출력하는 프로그램을 만듭니다. 문장은 한번만 입력하며 문자열 검색은 end가 입력될 때까지 반복합니다.

```
문장 입력  ifyou you would be loved, loooooove and be lovable
검색할 문자열 입력 (종료는 end) : be
검색 결과 : 2개
검색할 문자열 입력 (종료는 end) : ooo
검색 결과 : 2개
검색할 문자열 입력 (종료는 end) : oooo
검색 결과 : 1개
검색할 문자열 입력 (종료는 end) : lovely
검색 결과 : 0개
검색할 문자열 입력 (종료는 end) : end
```

실습 9 정답

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void)
5 {
6     char str[800];        // 문장을 입력할 배열
7     char ss[80];          // 검색할 문자열을 입력할 배열
8     int cnt = 0, i;       // cnt 찾은 문자열의 수를 누적할 변수, i는 반복 제어변수
9     int str_len, ss_len;  // 문장의 길이와 찾을 문자열의 길이를 저장할 변수
10
11     printf("문장 입력 : ");
12     fgets(str, sizeof(str), stdin);
13     str[strlen(str) - 1] = '\0'; // 새줄문자 제거
14     str_len = strlen(str);       // 입력된 문장의 길이 계산
15
16     while(1)
17     {
18         cnt = 0; // 누적 변수 초기화
19
20         printf("검색할 문자열 입력 (종료는 end) : ");
21         fgets(ss, sizeof(ss), stdin);
22         ss[strlen(ss) - 1] = '\0';
23         if(strcmp(ss, "end") == 0) break;
24         ss_len = strlen(ss);
25
26         for(i = 0; i < str_len; i++)
27         {
28             if((str_len - i) < ss_len) break;
29             if(strncmp(ss, str + i, ss_len) == 0)
30             {
31                 cnt++;
32                 i += (ss_len - 1);
33             }
34         }
35         printf("검색 결과 : %d개\n", cnt);
36     }
37     return 0;
38 }
```

```
문장 입력 ifyou you would be loved, loooooove and be lovable
검색할 문자열 입력 (종료는 end) : be
검색 결과 : 2개
검색할 문자열 입력 (종료는 end) : ooo
검색 결과 : 2개
검색할 문자열 입력 (종료는 end) : oooo
검색 결과 : 1개
검색할 문자열 입력 (종료는 end) : lovely
검색 결과 : 0개
검색할 문자열 입력 (종료는 end) : end
```

실습 10

문제 10)

문자열에서 특정 문자 변경 함수 만들기 StrReplace

```
void main( void )
{
    char string[100];

    strcpy( string, "It is a good time!" );
    StrReplace( string, 'i', 'I' );
    puts( string );
}
```

```
It Is a good tIme!
```

실습 10 정답

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char* StrReplace( char* pstring, char oldChar, char newChar );
5
6  void main( void )
7  {
8      char string[100];
9
10     strcpy( string, "It is a good time!" );
11     StrReplace( string, 'i', 'I' );
12     puts( string );
13 }
14 char* StrReplace( char* pstring, char oldChar, char newChar )
15 {
16     while( *pstring )
17     {
18         if( *pstring == oldChar )
19         {
20             *pstring = newChar;
21         }
22         pstring++;
23     }
24     return pstring;
25 }
```



It Is a good tIme!

실습 11

문제 11)

문자열 좌우에서 추출하기 (strdup 함수 사용)

StrLeft() 함수는 왼쪽에서 일정한 수만큼 추출

StrMid() 함수는 중간에서 끝까지 추출

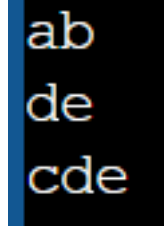
StrRight() 함수는 오른쪽에서 일정한 수만큼 추출

```
void main( void )
{
    char string[100];
    char* p;

    strcpy( string, "abcde" );
    p = StrLeft( string, 2 );
    puts( p ); free( p );

    strcpy( string, "abcde" );
    p = StrRight( string, 2 );
    puts( p ); free( p );

    strcpy( string, "abcde" );
    p = StrMid( string, 2 );
    puts( p ); free( p );
}
```



ab
de
cde

실습 11 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  char* StrLeft ( char* string, int len );
6  char* StrMid  ( char* string, int pos );
7  char* StrRight( char* string, int len );
8
9  void main( void )
10 {
11     char string[100];
12     char* p;
13
14     strcpy( string, "abcde" );
15     p = StrLeft( string, 2 );
16     puts( p ); free( p );
17
18     strcpy( string, "abcde" );
19     p = StrRight( string, 2 );
20     puts( p ); free( p );
21
22     strcpy( string, "abcde" );
23     p = StrMid( string, 2 );
24     puts( p ); free( p );
25 }
26 char* StrLeft ( char* string, int len )
27 {
28     char* p = strdup( string );
29     p[len] = 0;
30     return p;
31 }
32
33 char* StrMid( char* string, int pos )
34 {
35     return strdup( &string[pos] );
36 }
37
38 char* StrRight( char* string, int len )
39 {
40     int n = strlen(string);
41     return strdup( &string[n-len] );
42 }
```

ab
de
cde

실습 12

문제 12)

문자열을 회전하여 출력하기

```
[0] .String  
[1] g.Strin  
[2] ng.Stri  
[3] ing.Str  
[4] ring.St  
[5] tring.S  
[6] String.
```


실습 12 정답

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void main( void )
5 {
6     int i, j, len;
7     char ch;
8     char string[100];
9
10    strcpy( string, "String." );
11    len = strlen( string );
12
13    for( i=0; i<len; i++ )
14    {
15        ch = string[len-1];
16
17        for( j=len-1; j>0; j-- )
18        {
19            string[j] = string[j-1];
20        }
21        string[0] = ch;
22
23        printf( "[%d] %s \n", i, string );
24    }
25 }
```

```
✓ ↗ 🖨
[0] .String
[1] g.Strin
[2] ng.Stri
[3] ing.Str
[4] ring.St
[5] tring.S
[6] String.
```

질문 및 정리

