Chapter 11

문자열과 문자열 함수

Section 1 문자열 입출력
Section 2 문자열 함수
Section 3 문자열 정렬과 명령행 전달인자

무엇을 학습할까요?

- 문자열 입출력
- 문자열 함수
- 문자열 정렬과 명령행 전달인자
- 실전 프로젝트(연습문제 풀이, 총 16 문제)

Fig 11.1 문자열 배열의 초기화(p521)

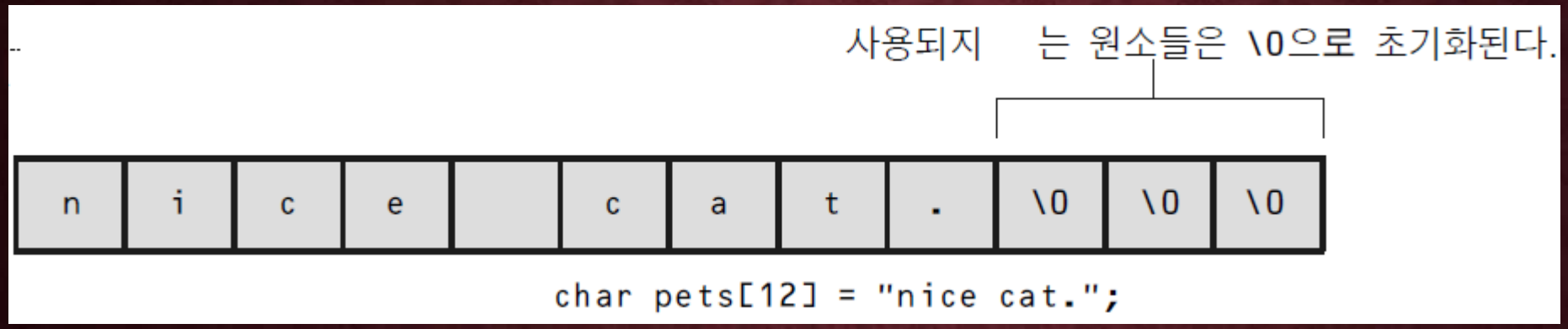
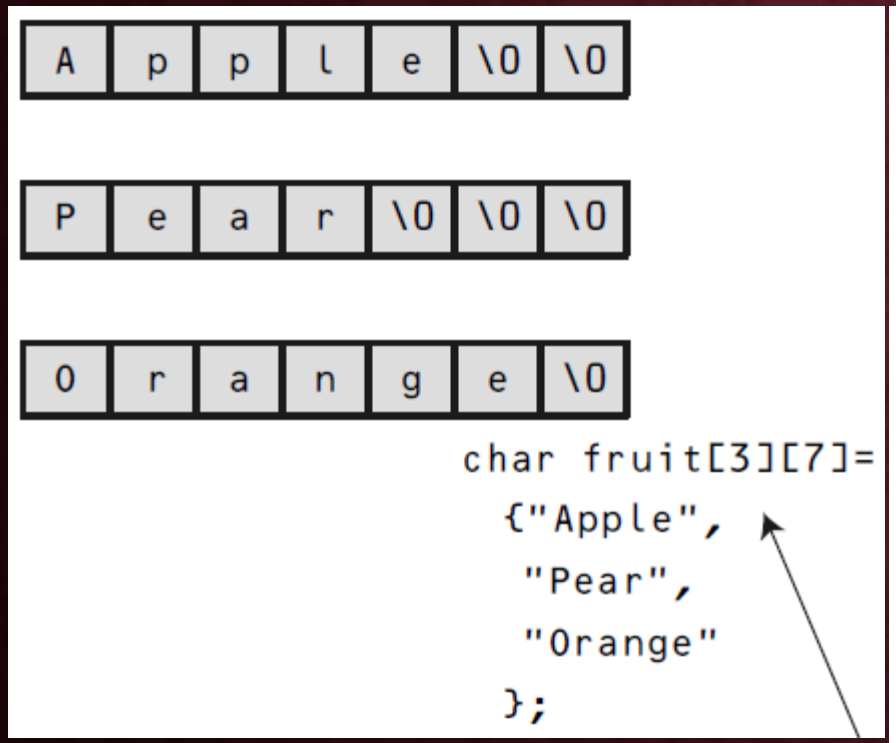


Fig 11.2 배열과 톱니형 배열(p529)



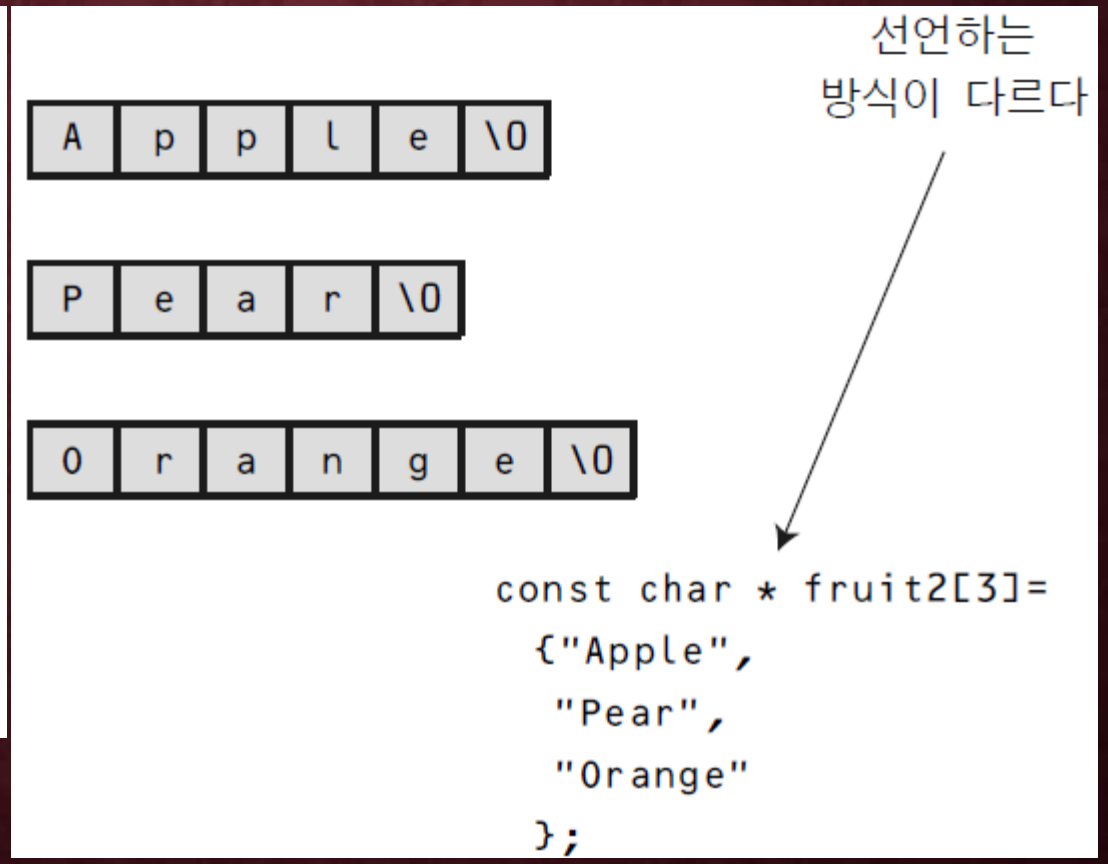


Fig 11.4 puts( ) 함수(p553)

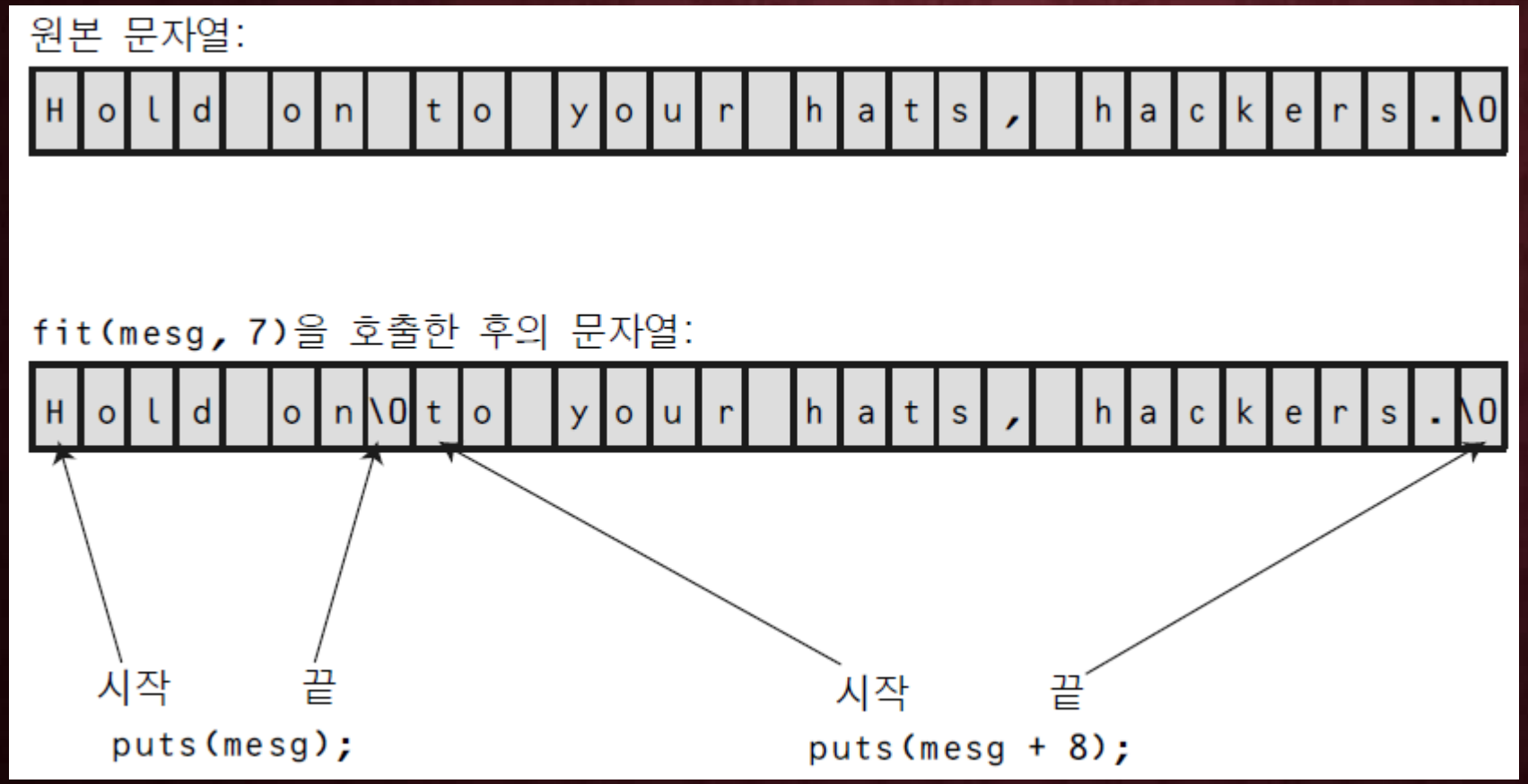


Fig 11.5 strcpy( ) 함수(p569)

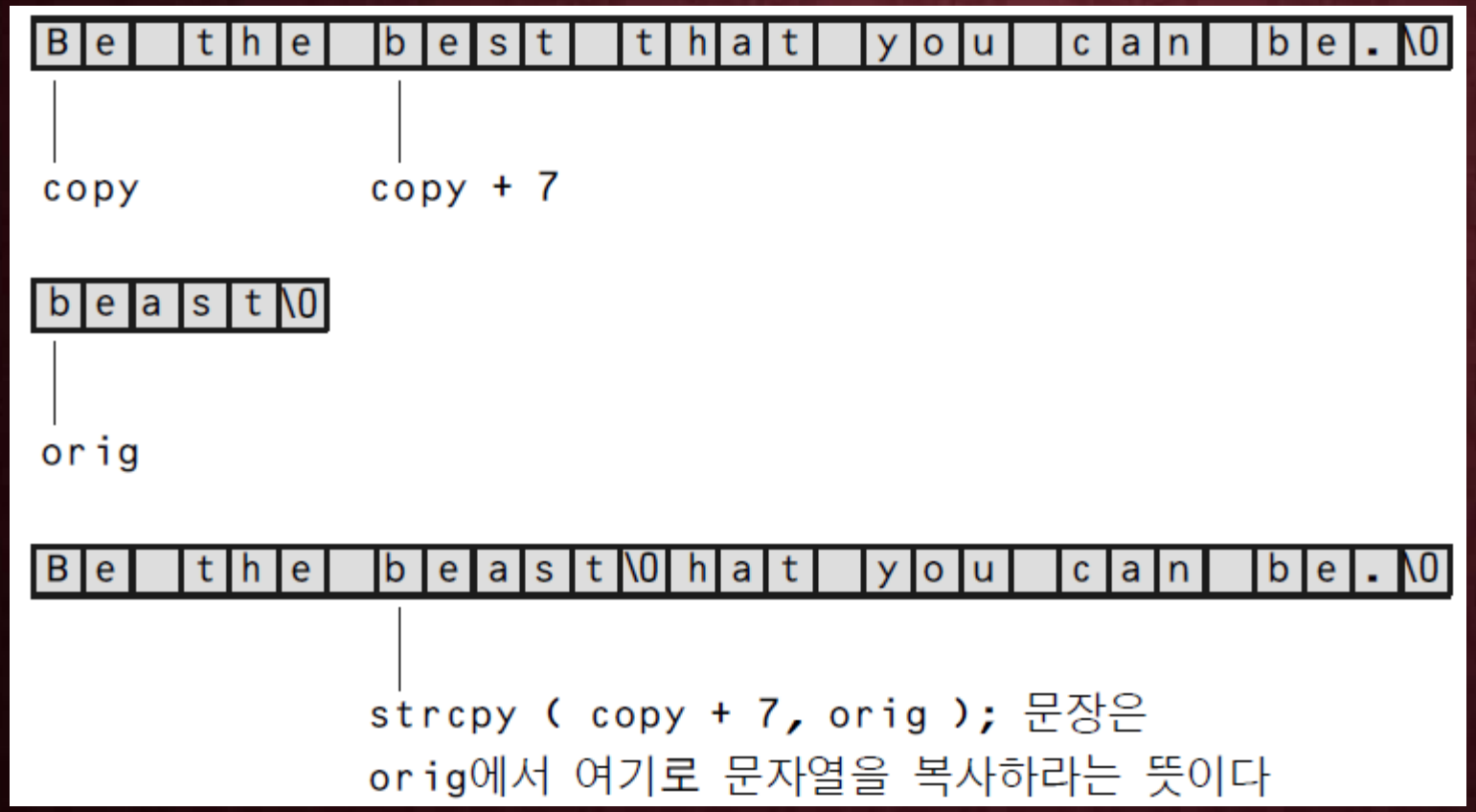


Fig 11.6 문자열 포인터들의 정렬(p521)

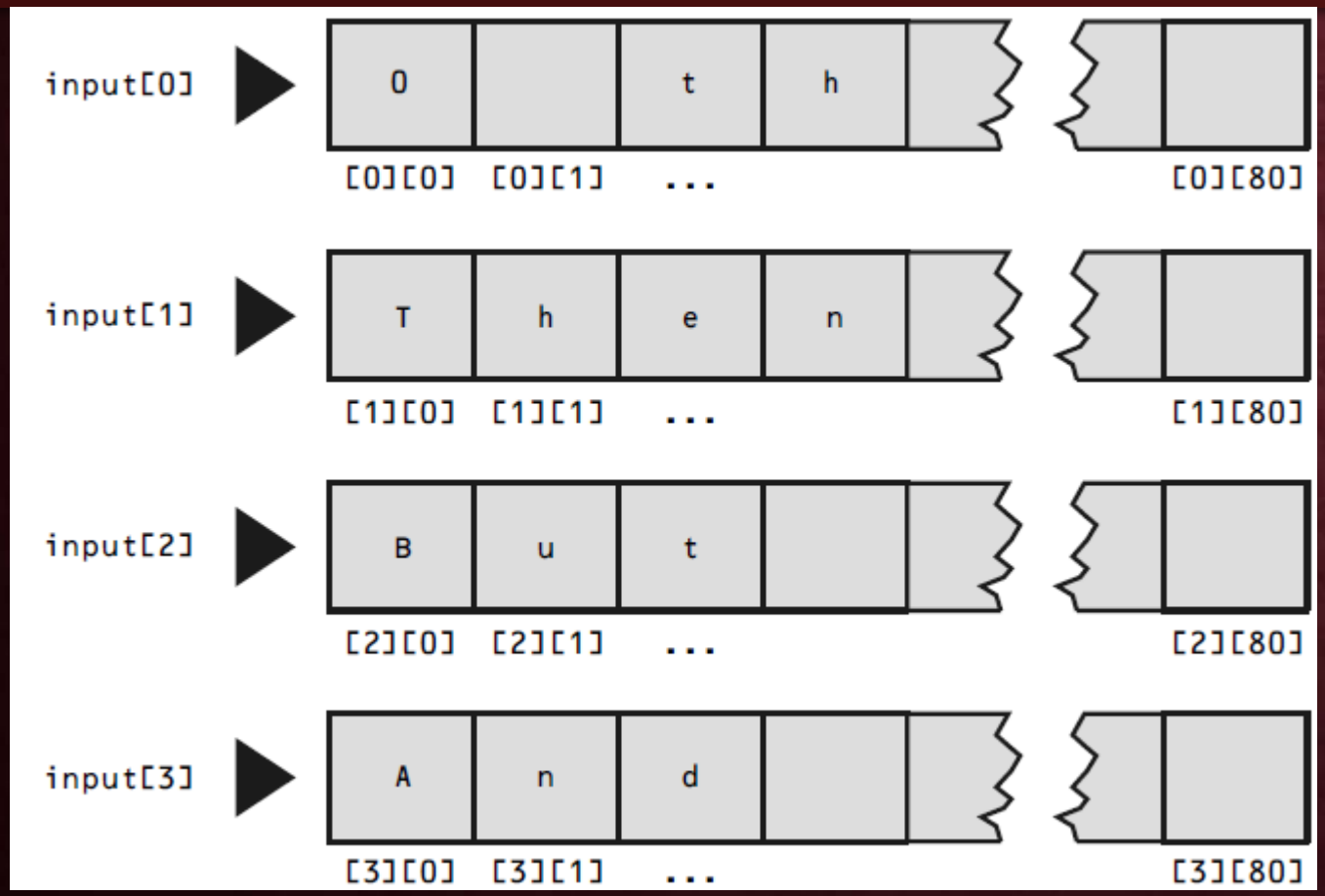


Fig 11.7 명령행 전달인자(p583)

repeat이라는 실행 파일

```
/*repeat*/
int main(int argc,char*argv[])
{
.
.
.
}
```

명령행에 다음과 같이 타이핑하여 프로그램을 실행한다

repeat I'm fine

argv[0] argv[1] argv[2]

명령행 전달인자들

argc = 3

문자열 3개

프로그래밍 연습 문제 p597

01 (공백(blank), 탭, 개행을 포함하여) 입력에서 다음 n개의 문자들을 읽고, 주소가 전달인자로 전달되는 배열에 그 결과를 저장하는 함수를 작성하고 테스트하라.

프로그래밍 연습 소스 01

```c
#include <stdio.h>
#define LEN 10
char * getnchar(char * str, int n);

int main(void)
{
    char input[LEN];
    char *check;
    check = getnchar(input, LEN - 1);
    if (check == NULL)
        puts("Input failed.");
    else
        puts(input);
    puts("Done.\n");
    return 0;
}
```

1

2

프로그래밍 연습 소스 01

```c
char * getnchar(char * str, int n)
{
    int i;
    int ch;
    for (i = 0; i < n; i++)
    {
        ch = getchar();
        if (ch != EOF)
            str[i] = ch;
        else
            break;
    }
    if (ch == EOF)
        return NULL;
    else
    {
        str[i] = '\0';
        return str;
    }
}
```

```
I can do it
I can do
Done.
```

프로그래밍 연습 문제 p597

02

n개의 문자들을 읽은 후에 또는 첫 번째 공백, 탭, 개행을 만난 후에
(어느 것이 먼저 일어나더라도) 읽기를 멈추도록 연습 ① 에서
작성한 함수를 수정하고 테스트하라.
(scanf()를 사용하지 마라.)

프로그래밍 연습 소스 02

```c
#include <stdio.h>
#define LEN 10
char * getnchar(char * str, int n);
int main(void)
{
    char input[LEN];
    char *check;
    check = getnchar(input, LEN - 1);
    if (check == NULL)
        puts("Input failed.");
    else
        puts(input);

    puts("Done.\n");
    return 0;
}
```

프로그래밍 연습 소스 02

```c
#include <ctype.h>
char * getnchar(char * str, int n)
{
    int i;
    int ch;
    for (i = 0; i < n; i++)
    {
        ch = getchar();
        if (ch == EOF || isspace(ch))
            break;
        else
            str[i] = ch;
    }
    if (ch == EOF)
        return NULL;
    else
    {
        str[i] = '\0';
        return str;
    }
}
```

1 2 3 4

```
I can do it.
I
Done.
```

프로그래밍 연습 문제 p597

03

한 라인의 입력에서 첫 번째 단어만 읽어 배열에 저장하고,
나머지는 버리는 함수를 작성하고 테스트하라
. 선두의 화이트 스페이스는 건너뛰어야 한다
. 여기서 단어는 스페이스, 탭, 개행이 들어 있지 않는 연속된 문자들을 말한다.

프로그래밍 연습 소스 03

```c
#include <stdio.h>
#define LEN 80
char * getword(char * str);
int main(void)
{
    char input[LEN];
    while (getword(input) != NULL)
        puts(input);
    puts("Done.\n");
    return 0;
}
```

프로그래밍 연습 소스 03

```
13 #include <ctype.h>
14 char * getword(char * str)
15 {
16     int ch;
17     char * orig = str;
18     // skip over initial whitespace
19     while ((ch = getchar()) != EOF && isspace(ch))
20         continue;
21
22     if (ch == EOF)
23         return NULL;
24     else
25         *str++ = ch; // first character in word
```

1 2 3 4

프로그래밍 연습 소스 03

```
27      // get rest of word
28      while ((ch = getchar()) != EOF && !isspace(ch))
29          *str++ = ch;
30      *str = '\0';  1
31
32      if (ch == EOF)
33          return NULL;
34      else
35      {                 2
36          while (ch != '\n')
37              ch = getchar();
38          return orig;
39      }
40  }
```

```
Vision can
Vision
^Z
Done.
```

프로그래밍 연습 문제

p597

번에 있는 설명을 참조하라.

04

프로그래밍 연습문제③ 에서 설명한 함수를,
읽을 수 있는 최대 문자수를 지정하는 제2 매개변수를
받아들이는 것을 제외하여 작성하고 테스트하라.

프로그래밍 연습 소스 04

```c
#include <stdio.h>
#define LEN 7
char * getword(char * str, int lim);

int main(void)
{
    char input[LEN];
    while (getword(input, LEN) != NULL && input[0] != '\0')
        puts(input);
    puts("Done.\n");
    return 0;
}
```

프로그래밍 연습 소스 04

```
15 char * getword(char * str, int lim)
16 {
17     int ch;
18     int ct = 0;
19     // dispose of poor choices
20     if (lim < 1)
21         return NULL;
22     else if (lim == 1)
23     {
24         str[0] = '\0';
25         return str;
26     }
```

프로그래밍 연습 소스 04

```c
    // skip over initial whitespace
    while ((ch = getchar()) != EOF && isspace(ch))
        continue;

    if (ch == EOF)
        return NULL;
    else
```

```
Vision can
Vision
^Z
Done.
```

```
36      // get rest of word
37      while (ct < lim - 1 && (ch = getchar()) != EOF && !isspace(ch))
38          str[ct++] = ch;
39      str[ct] = '\0';
40
41      if (ch == EOF)
42          return NULL;
43      else
44      {
45          while (ch != '\n')
46              ch = getchar();
47          return str;
48      }
49  }
```

witch를 사용하여 연습

04

프로그래밍 연습 문제 p597

를 다시 작성하라.

05

첫번째 함수 매개변수로 지정하는 문자열에서 두 번째 함수 매개변수로 지정하는 문자가 처음 나타나는 위치를 찾는 함수를 작성하고 테스트하라. 찾는 데 성공하면, 함수는 그 문자를 가리키는 포인터를 리턴한다. 그 문자열에서 그 문자를 찾지 못하면 함수는 널 포인터를 리턴한다. (이것은 라이브러리 함수 strchr()가 수행하는 기능과 같다.) 루프를 사용하여 함수에 입력 값들을 반복적으로 제공하는 완전한 프로그램을 만들어 작성한 함수를 테스트하라.

프로그래밍 연습 소스 05

```c
#include <stdio.h>
#include <string.h>
#define LEN 80
char * findchar(const char * str, char c);
char * s_gets(char * st, int n);
int main(void)
{
    char input[LEN];
    char ch;
    char * where;

    printf("Enter a string: ");
```

프로그래밍 연습 소스 05

```
13      while (s_gets(input, LEN) && input[0] != '\0')
14      {
15          printf("Enter a character: ");
16          ch = getchar();
17          while (getchar() != '\n')
18              continue;
19
20          where = findchar(input, ch);
21          if (where == NULL)
22              printf("%c not found in string.\n", ch);
23          else
24              printf("%c found at start of substring %s\n", ch, where);
25          printf("Next string: ");
26      }
27      puts("Done.\n");
28      return 0;
29  }
```

프로그래밍 연습 소스 05

```c
31 char * findchar(const char * str, char ch)
32 {
33     while (*str != ch && *str != '\0')
34         str++;
35
36     if (*str == '\0')
37         return NULL;
38     else
39         return (char *)str; /* cast const away */
40 }
```

```
Enter a string: I can do it.
Enter a character: can
c found at start of substring can do it.
Next string: k
Enter a character: a
a not found in string.
Next string:
Done.
```

프로그래밍 연습 소스 05

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;

    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else
            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

프로그래밍 연습 문제

p597

06

하나의 문자와 하나의 문자열 포인터를 두 개의 함수 매개변수로 사용하는 is_within() 함수를 작성하라.

그 함수는 문자열 안에 그 문자가 들어 있으면 0이 아닌 값(참)을 리턴하고, 그렇지 않으면 0(거짓)을 리턴한다.

루프를 사용하여 함수에 입력 값들을 반복적으로 제공하는 완전한 프로그램을 만들어 작성한 함수를 테스트하라.

프로그래밍 연습 소스 06

```c
#include <stdio.h>
#include <string.h>
#define LEN 80
#define _Bool int
_Bool is_within(const char * str, char c);
char * s_gets(char * st, int n);
int main(void)
{
    char input[LEN];
    char ch;
    int found;;
    printf("Enter a string: ");
```

프로그래밍 연습 소스 06

```
13      while (s_gets(input, LEN) && input[0] != '\0')
14      {
15          printf("Enter a character: ");
16          ch = getchar();
17          while (getchar() != '\n')
18              continue;
19          found = is_within(input, ch);
20          if (found == 0)
21              printf("%c not found in string.\n", ch);
22          else
23              printf("%c found in string %s\n", ch, input);
24          printf("Next string: ");
25      }
26      puts("Done.\n");
27      return 0;
28  }
```

프로그래밍 연습 소스 06

```c
_Bool is_within(const char * str, char ch)
{
    while (*str != ch && *str != '\0')
        str++;
    return *str; /* = 0 if \0 reached, non-zero otherwise */
}
```

프로그래밍 연습 소스 06

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;
    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline

        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else
            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

```
Enter a string: I can do it.
Enter a character: can
c found in string I can do it.
Next string:
Done.
```

프로그래밍 연습 문제

p598

07

strncpy(s1, s2, n) 함수는 s2에서 s1으로 정확히 n개의 문자들을 복사한다.
필요하면 s2를 자르거나 여분의 널 문자들로 채운다.
타깃 문자열은 s2의 길이가 n이거나 그 이상이면
널 문자로 종료되지 않을 수도 있다.
이 함수는 s1을 리턴한다.
이와 동일한 기능을 수행하는 함수를 직접 작성하고
mystrncpy()라고 부르라.
루프를 사용하여 함수에 입력 값들을 반복적으로 제공하는
완전한 프로그램을 만들어 작성한 함수를 테스트하라.

프로그래밍 연습 소스 07

```c
#include <stdio.h>
#define LEN 10
char * mystrncpy(char * s1, const char * s2, int n);
int main(void)
{
    char orig[LEN] = "transport";
    char copy[LEN];
    puts(orig);
    mystrncpy(copy, orig, 10);
    puts(copy);
    puts(mystrncpy(copy, orig, 10));
    mystrncpy(copy, orig, 4);
    copy[4] = '\0';
    puts(copy);
    puts("Done.\n");
    return 0;
}
```

프로그래밍 연습 소스 07

```c
char * mystrncpy(char * s1, const char * s2, int n)
{
    char * start = s1;
    while (n > 0 && *s2 != '\0')
    {
        *s1 = *s2;
        s1++;
        s2++;
        n--;
    }
    /* or, more compactly
    while (n-- > 0 && *s2)
    *s1++ = *s2++;
    */
    while (n-- > 0)
        *s1 = '\0';
    return start;
}
```

```
transport
transport
transport
tran
Done.
```

08

두 개의 문자열 포인터를 전달인자로 사용하는 함수 string_in()를 작성하라.
두 번째 문자열이 첫 번째 문자열 안에 있으면,
포함된 문자열이 시작되는 위치의 주소를 리턴한다.

예를 들어, string_in("hats", "at")은 hats에 있는 a의 주소를 리턴한다.
그렇지 않다면 널 포인터를 리턴한다.
루프를 사용하여 함수에 입력 값들을 반복적으로 제공하는
완전한 프로그램을 만들어 작성한 함수를 테스트하라.

프로그래밍 연습 소스 08

```c
#include <stdio.h>
#define LEN 20
char * string_in(const char * s1, const char * s2);
int main(void)
{
    char orig[LEN] = "transportation";
    char * find;
    puts(orig);
    find = string_in(orig, "port");
```

프로그래밍 연습 소스 08

```
11      if (find)
12          puts(find);
13      else
14          puts("Not found");
15      find = string_in(orig, "part");
16      if (find)
17          puts(find);
18      else
19          puts("Not found");
20      return 0;
21  }
```

```
transportation
portation
Not found
```

프로그래밍 연습 소스 08

```c
#include <string.h>
char * string_in(const char * s1, const char * s2)
{
    int l2 = strlen(s2);
    int tries; /* maximum number of comparisons */
    int nomatch = 1; /* set to 0 if match is found */
    tries = strlen(s1) + 1 - l2;
    if (tries > 0)
        while ((nomatch = strncmp(s1, s2, l2)) && tries--)
            s1++;
    if (nomatch)
        return NULL;
    else
        return (char *)s1; /* cast const away */
}
```

프로그래밍 연습 문제 p598

09

문자열의 내용을 거꾸로 뒤집은 문자열로 대체하는 함수를 작성하라. 루프를 사용하여 함수에 입력을 반복적으로 제공하는 완전한 프로그램을 만들어 작성한 함수를 테스트하라.

프로그래밍 연습 소스 09

```c
#include <stdio.h>
#include <string.h> // for strchr();
#define LEN 81
void string_rev(char * s);
char * s_gets(char * st, int n);
int main(void)
{
    char orig[LEN];
    puts("Enter a string of 80 characters or less:");
    while (s_gets(orig, LEN) && orig[0] != '\0')
    {
        string_rev(orig);
        puts(orig);
        puts("Enter next string (or just Enter to quit):");
    }
    puts("Bye!");
    return 0;
}
```

프로그래밍 연습 소스 09

```
20 /*
21 void reverse(char *str)
22 {
23     if(*(str+1) != '\0')
24         reverse(str+1); // 재귀 함수, 문자열의 맨끝이 아니면 계속
25
26     cout << *str;
27 }
28 */
```

프로그래밍 연습 소스 09

```c
#include <string.h>
void string_rev(char * s)
{
    int len = strlen(s);
    int i, j;
    char temp;
    for (i = 0, j = len - 1; i < j; i++, j--)
    {
        temp = s[i];
        s[i] = s[j];
        s[j] = temp;
    }
}
```

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;
    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else
            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

```
Enter a string of 80 characters or less:
I can do it.
.ti od nac I
Enter next string (or just Enter to quit):

Bye!
```

프로그래밍 연습 문제 p598

10

문자열을 전달인자로 취하여,
그 문자열에 있는 스페이스들을 모두 제거하는 함수를 작성하라.
빈 라인을 입력할 때까지 라인들을 반복하여 읽는 루프를 사용하는
프로그램을 작성하여 그 함수를 테스트하라.

프로그램은 그 함수를 각 입력 문자열에 적용하여,
그 결과를 표시해야 한다.

프로그래밍 연습 소스 10

```c
#include <stdio.h>
#include <string.h> // for strchr();
#define LEN 81
int drop_space(char * s);
char * s_gets(char * st, int n);
int main(void)
{
    char orig[LEN];
    puts("Enter a string of 80 characters or less:");
    while (s_gets(orig, LEN) && orig[0] != '\0')
    {
        drop_space(orig);
        puts(orig);
        puts("Enter next string (or just Enter to quit):");
    }
    puts("Bye!");
    return 0;
}
```

프로그래밍 연습 소스 10

```c
int drop_space(char * s)
{
    char * pos;
    while (*s) /* 현재 포인터 값이 있는 동안 반복 or while (*s != '\0') */
    {
        if (*s == ' ') /* 현재 포인터(위치)가 공백인지? */
        {
            pos = s;
            do
            {
                *pos = *(pos + 1);
                pos++; // 공백인 경우 포인터(위치) 한 칸 증가
            } while (*pos);
        }
        else
            s++;
    }
}
```

프로그래밍 연습 소스 10

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;
    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else
            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

```
Enter a string of 80 characters or less:
I can do it.
Icandoit.
Enter next string (or just Enter to quit):

Bye!
```

프로그래밍 연습 문제 p598

11

문자열을 10개까지 또는 EOF를 만날 때까지 읽는 프로그램을 작성하라.
둘 중 어느 것이 먼저 일어나더라도 읽기를 멈춘다.
이 프로그램은 사용자가 메뉴로 다음과 같은 5가지 선택권을 제공해야 한다:
문자열들의 원본 리스트를 출력한다,
문자열들을 ASCII 조회 순서로 출력한다,
문자열들을 길이가 커지는 순서로 출력한다,
문자열들을 첫 단어의 길이가 커지는 순서로 출력한다.
프로그램을 종료한다.
이 프로그램은 사용자가 종료를 선택할 때까지 계속 실행되어야 한다.
물론 이 프로그램은 메뉴에 제시된 작업을 실제로 수행해야 한다.

프로그래밍 연습 소스 11

```c
#include <stdio.h>
#include <string.h>
#define LEN 81
#define STRS 10

void menu(void);
void orig_order(char * p[], const char car[][LEN], int n);
void ascii_order(char * p[], int n);
void length_order(char * p[], int n);
void len1st_order(char * p[], int n);
int word_len(const char * s);
char * s_gets(char * st, int n);
```

프로그래밍 연습 소스 11

```c
int main(void)
{
    char orig[STRS][LEN];
    char * ptrs[STRS];
    int ct = 0;
    int i;
    char choice;

    printf("Enter up to %d lines of text ", STRS);
    printf("or just press Enter to quit):\n");
    while (ct < STRS && s_gets(orig[ct], LEN) && orig[ct][0] != '\0')
    {
        ptrs[ct] = orig[ct];
        ct++;
    }
```

프로그래밍 연습 소스 11

```
30      while ((choice = getchar()) != 'q')
31      {
32          if (choice == '\n')
33              continue;
```

프로그래밍 연습 소스 11

```
34          switch (choice)
35          {
36          case 'o': orig_order(ptrs, orig, ct);
37              break;
38          case 'a': ascii_order(ptrs, ct);
39              break;
40          case 'l': length_order(ptrs, ct);
41              break;
42          case 'w': len1st_order(ptrs, ct);
43              break;
44          default: printf("%c not a valid choice; try again.\n", choice);
45              menu();
46              continue;
47          }
```

프로그래밍 연습 소스 11

```
48          for (i = 0; i < ct; i++)
49              puts(ptrs[i]);
50          menu();
51      }
52      puts("Bye!");
53      return 0;
54  }
```

프로그래밍 연습 소스 11

```c
void menu(void)
{
    puts("Enter o, a, l, w, or q:");
    puts("o) original order a) ASCII order");
    puts("l) string-length order w) length of first word order");
    puts("q) quit");
}

void orig_order(char * p[], const char car[][LEN], int n)
{
    int i;
    for (i = 0; i < n; i++)
        p[i] = (char *)car[i]; /* car const, but p isn't */
}
```

프로그래밍 연습 소스 11

```
71 void ascii_order(char * p[], int n)
72 {
73     int top, comp;
74     char * temp;
75     for (top = 0; top < n - 1; top++)
76     {
77         for (comp = top + 1; comp < n; comp++)
78         {
79          1 if (strcmp(p[top], p[comp]) > 0)
80             {
81                 temp = p[top];
82                 p[top] = p[comp];
83                 p[comp] = temp;
84             }
85         }
86     }
87 }
```

프로그래밍 연습 소스 11

```
89  void length_order(char * p[], int n)
90  {
91      int top, comp;
92      char * temp;
93      for (top = 0; top < n - 1; top++)
94      {
95          for (comp = top + 1; comp < n; comp++)
96          {
97              if (strlen(p[top]) > strlen(p[comp]))
98              {
99                  temp = p[top];
100                 p[top] = p[comp];
101                 p[comp] = temp;
102             }
103         }
104     }
105 }
```

1

프로그래밍 연습 소스 11

```
107 int word_len(const char * s)
108 {
109     const char * start = s;
110     while (*s != ' ' && *s != '\0')
111     {
112         s++;
113     }
114     return s - start;
115 }
```

프로그래밍 연습 소스 11

```
117 void len1st_order(char * p[], int n)
118 {
119     int top, comp;
120     char * temp;
121     for (top = 0; top < n - 1; top++)
122     {
123         for (comp = top + 1; comp < n; comp++)
124         {
125           1 if (word_len(p[top]) > word_len(p[comp]))
126             {
127                 temp = p[top];
128                 p[top] = p[comp];
129                 p[comp] = temp;
130             }
131         }
132     }
133 }
```

프로그래밍 연습 소스 11

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;
    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else
            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

프로그래밍 연습 소스 11

```
Enter up to 10 lines of text or just press Enter to quit):
I can do it.
Vision
Dream
Happy

Enter o, a, l, w, or q:
o) original order a) ASCII order
l) string-length order w) length of first word order
q) quit
a
Dream
Happy
I can do it.
Vision
Enter o, a, l, w, or q:
o) original order a) ASCII order
l) string-length order w) length of first word order
q) quit
q
Bye!
```

프로그래밍 연습 문제 p599

12

EOF를 만날 때까지 입력을 읽어서 단어 수, 대문자 수, 소문자 수, 구두점 수, 숫자 수를 보고하는 프로그램을 작성하라.

ctype.h 계열의 함수들을 사용하라.

프로그래밍 연습 소스 12

```c
#include <stdio.h>
#include <ctype.h> // for isspace()
//#include <stdbool.h> // for bool, true, false
#define bool int
#define true 1
#define false 0
int main(void)
{
    char c; // read in character
    int low_ct = 0; // number of lowercase characters
    int up_ct = 0; // number of uppercase characters
    int dig_ct = 0; // number of digits
    int n_words = 0; // number of words
    int punc_ct = 0; // number of punctuation marks
    bool inword = false; // == true if c is in a word
    printf("Enter text to be analyzed (EOF to terminate):\n");
```

프로그래밍 연습 소스 12

```
17      while ((c = getchar()) != EOF)
18      {
19          if (islower(c))  1
20              low_ct++;
21          else if (isupper(c))  2
22              up_ct++;
23          else if (isdigit(c))  3
24              dig_ct++;
25          else if (ispunct(c))  4
26              punc_ct++;
27          if (!isspace(c) && !inword)
28          {
29              inword = true; // starting a new word
30              n_words++; // count word
31          }
32          if (isspace(c) && inword)
33              inword = false; // reached end of word
34      }
```

프로그래밍 연습 소스 12

```c
    printf("\nwords = %d, lowercase = %d, uppercase = %d, "
        "digits = %d, punctuation = %d\n",
        n_words, low_ct, up_ct, dig_ct, punc_ct);
    return 0;
}
```

```
Enter text to be analyzed (EOF to terminate):
I can do it.
^Z

words = 4, lowercase = 7, uppercase = 1, digits = 0, punctuation = 1
```

프로그래밍 연습 문제 p599

13

명령행 전달인자들을 역순으로 에코하는 프로그램을 작성하라. 예를 들어, 명령행 전달인자들이 see you later로 주어졌다면, 프로그램은 later you see라고 출력해야 한다.

프로그래밍 연습 소스 13

```c
#include <stdio.h>
int main(int argc, char *argv[])
{
    int ct;
    for (ct = argc - 1; ct > 0; ct--)
        printf("%s ", argv[ct]);
    putchar('\n');
    puts("Done");
    return 0;
}
```

```
C:₩Users₩Administrator>cd C:₩work₩c₩pe₩Debug

C:₩work₩c₩pe₩Debug>pe see you later
later you see
Done
```

프로그래밍 연습 문제 p599

14 명령행으로 동작시키는 멱승법(power-law) 프로그램을 작성하라. 명령행의 첫 번째 전달인자는 어떤 멱승을 구할 double형 밑수가 되고, 두 번째 전달인자는 정수형 지수가 되어야 한다.

프로그래밍 연습 소스 14

```c
#include <stdio.h>
#include <stdlib.h> /* for atof() */
#include <math.h> /* for pow() */
int main(int argc, char *argv[])
{
    double num, exp;
    if (argc != 3)
        printf("Usage: %s number exponent\n", argv[0]);
    else
    {
        num = atof(argv[1]);
        exp = atof(argv[2]);
        printf("%f to the %f power = %g\n", num, exp, pow(num, exp));
    }
}
```

```
C:₩work₩c₩pe₩Debug>cd C:₩work₩c₩pe₩Debug

C:₩work₩c₩pe₩Debug>pe 2.5 3
2.500000 to the 3.000000 power = 15.625
```

프로그래밍 연습 문제 p599

15

문자 분류 함수들을 사용하여 atoi()와 동일한 기능을 수행하는 함수를 작성하라.

이 버전의 경우 입력 문자열이 순수한 수가 아니라면

0의 값을 반환하도록 하라.

프로그래밍 연습 소스 15

```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>
char * s_gets(char * st, int n);
int my_atoi(const char * str);
#define LIM 15
int main()
{
    char temp[LIM];
    puts("Enter an integer (empty line to quit):");
    while (s_gets(temp, LIM) && temp[0] != '\0')
    {
        printf("%s: %d\n", temp, my_atoi(temp));
        puts("Enter an integer (empty line to quit):");
    }
    printf("Bye!\n");
    return 0;
}
```

프로그래밍 연습 소스 15

```
20 int my_atoi(const char * str)
21 {
22     int sign = 1;
23     int value = 0; // 음수와 양수를 구별하기 위한 변수 (기본값 양수)
24
25     while (isspace(*str))  // 문자열 앞의 공백 처리
26         str++;
27
28     if (*str == '-') // 부호 처리
29     {
30         sign = -1;
31         str++;
32     }
33     else if (*str == '+')
34         str++;
```

프로그래밍 연습 소스 15

```
36      while (*str != '\0' && isdigit(*str))  // 정수값 추출
37      {
38          value *= 10;
39          value += *str - '0';
40          str++;
41      }
42      if (*str)
43          value = 0; /* return 0 if string is not a pure number */
44      return sign * value;
45  }
```

프로그래밍 연습 소스 15

```c
char * s_gets(char * st, int n)
{
    char * ret_val;
    char * find;
    ret_val = fgets(st, n, stdin);
    if (ret_val)
    {
        find = strchr(st, '\n'); // look for newline
        if (find) // if the address is not NULL,
            *find = '\0'; // place a null character there
        else

            while (getchar() != '\n')
                continue;
    }
    return ret_val;
}
```

```
Enter an integer (empty line to quit):
123456
123456: 123456
Enter an integer (empty line to quit):

Bye!
```

16

파일 끝에 도달할 때까지 입력을 읽고, 그것을 디스플레이에 에코하는 프로그램을 작성하라. 프로그램은 다음과 같은 명령행 전달인자들을 인식하고 그 기능을 구현해야 한다.

-p	입력을 그대로 출력한다
-u	입력을 모두 대문자로 변경하여 출력한다
-l	입력을 모두 소문자로 변경하여 출력한다

또한 명령행 전달인자가 없다면,
프로그램이 -p 전달인자와 사용된 것처럼 실행되게 하라.

프로그래밍 연습 소스 16

```c
#include <stdio.h>
#include <ctype.h>
int main(int argc, char *argv[])
{
    char mode = 'p';
    int ok = 1;
    int ch;
    if (argc > 2)
    {
        printf("Usage: %s [-p | -u | -l]\n", argv[0]);
        ok = 0; /* skip processing input */
    }
```

프로그래밍 연습 소스 16

```c
13      else if (argc == 2)
14      {
15          if (argv[1][0] != '-')
16          {
17              printf("Usage: %s [-p | -u | -l]\n", argv[0]);
18              ok = 0;
19          }
20          else
21              switch (argv[1][1])
22          {
23              case 'p':
24              case 'u':
25              case 'l': mode = argv[1][1];
26                  break;
27              default: printf("%s is an invalid flag; ", argv[1]);
28                  printf("using default flag (-p).\n");
29          }
30      }
```

```
I can do it.
I can do it.
^Z
```

```
C:₩work₩c₩pe₩Debug>cd C:₩work₩c₩pe₩Debug

C:₩work₩c₩pe₩Debug>pe -u
I can do it.
I CAN DO IT.
^Z
```

프로그래밍 연습 소스 16

```
32      if (ok)
33          while ((ch = getchar()) != EOF)
34          {
35              switch (mode)
36              {
37              case 'p': putchar(ch);
38                  break;
39              case 'u': putchar(toupper(ch));
40                  break;
41              case 'l': putchar(tolower(ch));
42              }
43          }
44      return 0;
45  }
```

KeyPoint 핵심 개념

```
char fruit[3][7] = {"Apple", "Pear", " Orange"};
char *fruit[3] = {"Apple", "Pear", " Orange"};
char words[81];
gets(words); // 개행문자('n') 제거
fgets(words, 81, stdout); // 개행문자('n') 포함
strlen(words); strcat(words, "Concatenation");
strcmp(a, b);, strcpy(dest, source);
```

이번 학습에 이어!

- 다음 강의 내용
  - 제 12장 저장 클래스, 연계, 메모리 관리, 연습문제

Thank you !