

## CHAPTER 2

# 파일 입출력의 이해 파일 처리 : 텍스트 파일과 이진 파일 실습

# 실습 1

## 문제 1)

프로그램상에서 mystory.txt 라는 이름의 파일을 생성해서 본인의 이름, 주민번호, 전화번호를 저장하는 프로그램을 작성하시오.

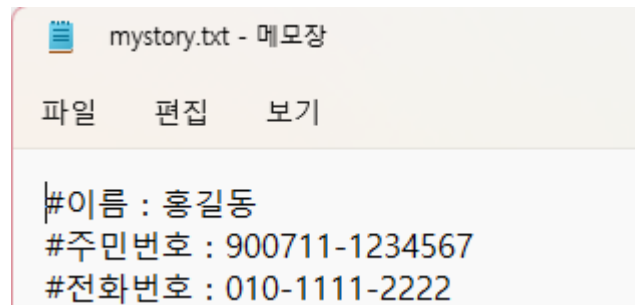
입력의 형태는 아래와 같다.

#이름 : 홍길동

#주민번호 : 900711-1234567

#전화번호 : 010-1111-2222

(출력이 되면 메모장으로 확인이 가능해야 한다.)



# 실습 1 정답

```
/*
 * 문제1. 프로그램상에서 mystory.txt 라는 이름의 파일을 생성해서
 * 본인의 이름, 주민번호, 전화번호를 저장하는 프로그램을 작성하자
 * 입력의 형태는 아래와 같다.
 *
 * #이름 : 홍길동
 * #주민번호 : 900711-1234567
 * #전화번호 : 010-1111-2222
 *
 * (입력이 완성되면 메모장으로 확인이 가능해야 한다.)
 */
```

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

```
int main() {

    // 스트림 형성
    FILE* fw = fopen("mystory.txt", "w");
    FILE* fa = fopen("mystory.txt", "a");
    FILE* fr = fopen("mystory.txt", "r");

    ////////// 문제1 //////////
    // 문자열 입력
    fputs("#이름 : 홍길동\n", fw);
    fputs("#주민번호 : 900711-1234567\n", fw);
    fputs("#전화번호 : 010-1111-2222\n", fw);
    fclose(fw);
    ///////////////////////////////////
```

## 실습 2

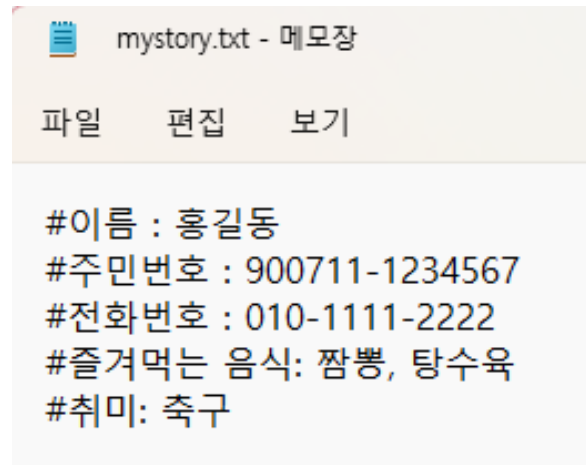
문제 2)

문제 1에서 작성한 파일에 데이터를 추가하시오.  
추가할 데이터는 즐겨먹는 음식의 정보와 취미이다.

입력의 형태는 다음과 같다.

#즐거먹는 음식 : 짬뽕, 탕수육

#취미 : 축구



## 실습 2 정답

```
/*
 * 문제2. 문제 1에서 작성한 파일에 데이터를 추가하자.
 * 추가할 데이터는 즐겨먹는 음식의 정보와 취미이다.
 * 입력의 형태는 다음과 같다.
 *
 * #즐거먹는 음식 : 짬뽕, 탕수육
 * #취미 : 축구
 *
 * (입력에 완성되면 메모장으로 확인이 가능해야 한다.)
 */
```

```
47
48 ////////////////////////////////////////////////// 문제2 ///////////////////////////////////
49 // 문자열 추가(수정)
50 fputs("#즐거먹는 음식: 짬뽕, 탕수육\n", fa);
51 fputs("#취미: 축구\n", fa);
52 fclose(fa);
53 ////////////////////////////////////////////
54
```

## 실습 3

문제 3)

문제 1과 문제 2에서 생성한 파일에 저장된 정보 전체를 파일로 입력(mystory.txt)받아 화면에 출력하는 하는 프로그램을 작성하시오.

```
Microsoft Visual Studio 디버그 콘솔
#이름 : 홍길동
#주민번호 : 900711-1234567
#전화번호 : 010-1111-2222
#즐거먹 음식: 짬뽕, 탕수육
#취미: 축구
```

# 실습 3 정답

```
/*
 * 문제3. 문제1과 문제2에서 생성한 파일에 저장된 정보 전체를 출력하는 프로그램을 작성하자.
 */
```

```
55
56 ////////////// 문제3 ///////////
57 // 파일을 읽어와 문장을 차례로 출력
58 char str[100];
59 for (int i = 0; i < 5; i++) {
60     fgets(str, sizeof(str), fr);
61     printf("%s", str);
62 }
63 fclose(fr);
64 ///////////////
65
66
67 return 0;
68 }
```

## 실습 4

문제 4)  
검색 프로그램을 작성하시오.

프로그램 사용자가 검색의 대상을 입력하면, 문제 1, 2, 3의 mystory.txt에서 검색하여 해당 행을 출력한다.

```
Microsoft Visual Studio 디버그 콘솔
찾으실 정보를 입력하세요 : 전화번호
문자열을 찾았습니다.
#전화번호 : 010-1111-2222
```



# 실습 4 정답

C: > Users > jcn99 > Desktop > work > 광운대초빙교원 > 고급C프로그래밍 > 강의교안 > 9주차 > 파일입출력2 > C언어실습\_9-4.c

```
1  /*
2  *  문제4.  검색 프로그램을 작성하자.
3  *          프로그램 사용자가 검색의 대상을 입력하면, 문제 1, 2, 3의 mystory.txt에서 검색하여
4  *          해당 행을 출력한다.
5  *          검색 예시는 아래와 같다.
6  *
7  *          (입력)
8  *          >> 찾으실 정보를 입력하세요 : 전화번호
9  *
10 *          (출력)
11 *          >> 문자열을 찾았습니다.
12 *          >> #전화번호 : 010-1111-2222
13 */
14
15 #define _CRT_SECURE_NO_WARNINGS
16 #include <stdio.h>
17 #define STR_SIZE    100
18
```

# 실습 4 정답

```
int main() {

    char ans[20];
    char str[STR_SIZE];
    int pass = 0;

    printf("찾으실 정보를 입력하세요 : ");
    scanf("%s", ans);

    // 스트림 형성
    FILE* fr = fopen("mystory.txt", "r");
    if (fr == NULL) {
        printf("파일 읽기 실패!");
        return -1;
    }

    // 문자열 검색
    while (fgets(str, STR_SIZE, fr) != NULL) { // 한 행씩 입력

        if (strstr(str, ans) != NULL) { // str에 ans가 있다면 (substring의
                                        // 첫 번째 주소 반환)
            printf("문자열을 찾았습니다. \n");
            printf("%s", str); // 찾은 행 출력
            pass++;           // 확인용
        }
    }

    // 한번도 확인되지 않은 경우
    if (pass == 0) printf("문자열을 찾을 수 없습니다.");
    fclose(fr);

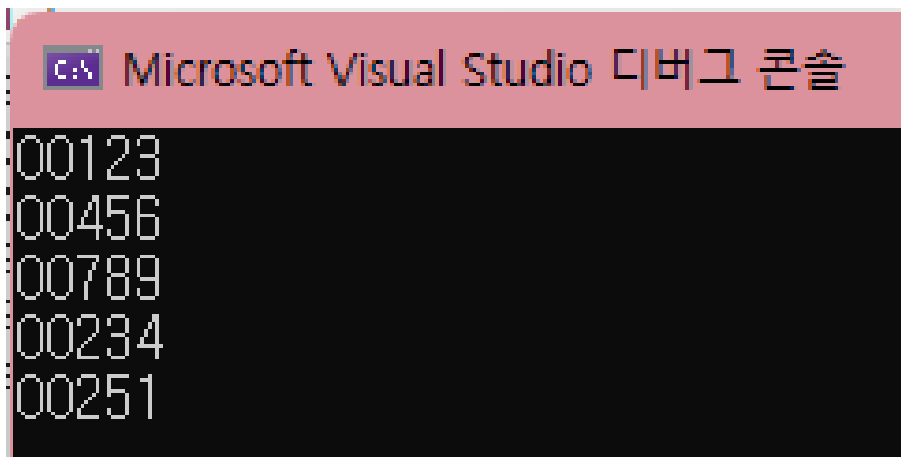
    return 0;
}
```

## 실습 5

문제 5)

int형 배열 arr을 코드 상에서 123, 456, 789, 234, 251로 초기화하고, 텍스트 파일로 저장하는 프로그램을 fprintf()를 사용하여 작성하시오.

또, fscanf()를 사용하여 파일로 저장된 각 데이터를 5자리의 출력 자릿수로 읽어 화면에 출력하시오.



```
C:\> Microsoft Visual Studio 디버그 콘솔
00123
00456
00789
00234
00251
```

# 실습 5 정답

seis > jch99 > Desktop > work > 경원대학교 > 교습 > C프로그래밍 > 강의표준 > 9주차 > 파일입출력2 > C언어실습\_9-5.c

```
/*
 * 문제 5. int형 배열 arr을 123,456,789,234,251로 초기화하고, 텍스트 파일로 저장하는 프로그램을 fprintf()를 사용하여 작성하시오.
 * 이때 fscanf()를 사용하여 파일로 저장된 각 데이터를 5자리의 출력 자릿수로 읽어 화면에 출력하시오.
 *
 * (출력)
 * >> 00123
 * >> 00456
 * >> 00789
 * >> 00234
 * >> 00251
 */

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#define ARR_NUM 5
```

# 실습 5 정답

```
13
14 #define _CRT_SECURE_NO_WARNINGS
15
16 #include <stdio.h>
17 #define ARR_NUM 5
18
19 int main() {
20
21     int arr[ARR_NUM] = { 123,456,789,234,251 };    // 배열 초기화. 입력받을 숫자들임.
22     int cpy[ARR_NUM] = { 0 };
23
24     // 스트림 생성
25     FILE* fw = fopen("arr.txt", "w");
26     FILE* fr = fopen("arr.txt", "r");
27
28     for (int i = 0; i < ARR_NUM; i++) {
29         fprintf(fw, "%d\n", arr[i]);                // 배열 파일에 출력
30     }
31     fclose(fw);                                     // 스트림 소멸
32
33     for (int i = 0; i < ARR_NUM; i++) {
34         fscanf(fr, "%d", &cpy[i]);                  // 파일에 있는 숫자 입력
35         printf("%05d \n", cpy[i]);                  // 복사된 숫자 출력
36     }
37     fclose(fr);                                     // 스트림 소멸
38 }
```

## 실습 6

문제 6)

텍스트 파일을 입력받아 파일 내의 모든 문자를 소문자로 변환해서 화면에 출력하시오.

또, 소문자로 변환된 문장을 새로운 파일로 출력하는 프로그램을 작성하시오.

입력받을 파일의 내용은 아래와 같다.

1. This Is Text File For Input Test.
2. THIS FILE CONTAINS MULTIPLE LINES OF TEXT.
3. the program displays the content of fiel with line number.

Microsoft Visual Studio 디버그 콘솔

```
1. this is text file for input test.  
2. this file contains multiple lines of text.  
3. the program displays the content of fiel with line number.
```

readyou.txt - 메모장

파일 편집 보기

```
1. this is text file for input test.  
2. this file contains multiple lines of text.  
3. the program displays the content of fiel with line numb
```

# 실습 6 정답

문제 6. 텍스트 파일을 입력받아 파일 내의 모든 문자를 소문자로 변환해서 화면에 출력하시오.  
또, 소문자로 변환된 문장을 새로운 파일로 출력하는 프로그램을 작성하시오.

(입력 예시)

파일 이름 : readme.txt

파일 내용 : 1. This Is Text File.

2. THIS FILE IS A MIXTURE OF UPPERCASE AND LOWERCASE LETTERS.

3. this program prints out the file.

(출력 예시)

파일 이름 : readyou.txt

파일 내용 : 1. this is text file.

2. this file is a mixture of uppercase and lowercase letters.

3. this program prints out the file.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

# 실습 6 정답

```
18
19 #define _CRT_SECURE_NO_WARNINGS
20 #include <stdio.h>
21
22 ∨ int main() {
23     // 스트림 생성
24     FILE* fr = fopen("readme.txt", "r");
25     FILE* fw = fopen("readyou.txt", "w");
26
27     char str[100];
28
29     while (fgets(str, sizeof(str), fr) != 0) { // 한 줄씩 입력받음.
30         for (int j = 0; j < 100; j++)
31             if (str[j] >= 'A' && str[j] <= 'Z') str[j] += 32; // 대문자에 해당하는 경우만 소문자로 치환.
32         printf("%s", str); // 화면에 출력
33         fputs(str, fw); // 파일에 출력
34     }
35
36     fclose(fw);
37     fclose(fr);
38
39     return 0;
40 }
41
42
43
```

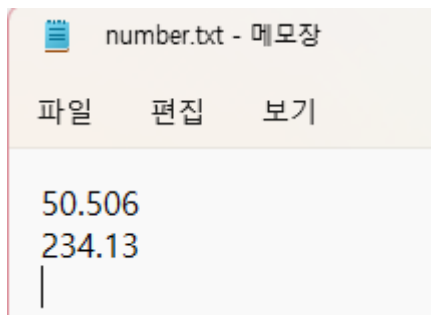


# 실습 7

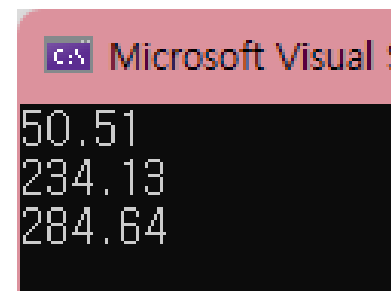
문제 1)

실수 2개를 파일로 입력 받아 합계를 구하고 각각 소수 셋째 자리에서 반올림하여 둘째 자리까지 개행하여 출력하시오.

(입력)



(출력)



# 실습 7 정답

```
C: > Users > jcn99 > Desktop > work > 광운대초방교원 > 고급C프로그래밍 > 강의교안 > 9주차 > 파일입  
1  /*  
2  *      [문제 7]  
3  *      실수 2개를 '파일로 입력' 받아 합계를 구하고 각각 소수  
4  *      셋째 자리에서 반올림하여 둘째 자리까지 대행하여 '화면에 출력'하시오.  
5  *  
6  *      (파일 입력)  
7  *      파일 이름 : number.txt  
8  *      파일 내용 : 50.506  
9  *                  234.13  
10 *  
11 *      (화면 출력)  
12 *      50.51  
13 *      234.13  
14 *      284.64  
15 *  
16 */
```

# 실습 7 정답

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

int main() {
    FILE* fr = fopen("number.txt", "r");

    double num = 0;
    double total = 0;

    while (fscanf_s(fr, "%lf", &num) != EOF) {
        printf("%.2lf \n", num);
        total += num;
    }

    printf("%.2lf \n", total);
    fclose(fr);

    return 0;
}
```

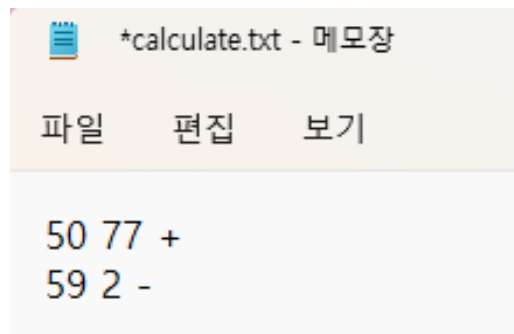
// 파일에 데이터가 있다면 개행하며 차례로 받음.  
// 받은 행 출력  
// total에 더함

## 실습 8

문제 8)

두 정수와 한 개의 연산자(+, -, \*, /, %)를 파일로 입력받아 입력 순서대로 계산하는 계산식을 화면으로 출력하시오.

(입력)

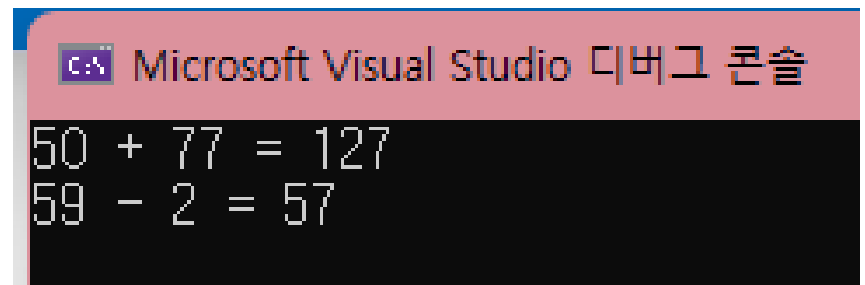


```
*calculate.txt - 메모장

파일  편집  보기

50 77 +
59 2 -
```

(출력)



```
Microsoft Visual Studio 디버그 콘솔

50 + 77 = 127
59 - 2 = 57
```

# 실습 8 정답

> Users > jch99 > Desktop > work > 평문대조명교원 > 고급C프로그래밍 > 강의교안 > 9주차 > 파일입출력2 > C언어실습\_9-8.c

```
1  /*
2  *      [문제 8]
3  *      두 정수와 한 개의 연산자(+, -, *, /, %)를 '파일로 입력'받아 '입력 순서대로 계산하는 계산식을 '화면으로 출력'하시오.
4  *
5  *      (파일 입력)
6  *      50 77 +
7  *      59 2 -
8  *
9  *      (화면 출력)
10 *      50 + 77 = 127
11 *      59 - 2 = 57
12 */
13
14
15 #define _CRT_SECURE_NO_WARNINGS
16 #include <stdio.h>
17
```

# 실습 8 정답

```
int main() {
    FILE* fr = fopen("calculate.txt", "r");

    int num1, num2, total = 0;
    char c = 0;

    // 파일에 데이터가 있다면
    while (fscanf_s(fr, "%d %d %c", &num1, &num2, &c) != EOF) {

        // 기호 확인 후 연산
        switch (c) {
            case '+':
                total = num1 + num2;
                break;
            case '-':
                total = num1 - num2;
                break;
            case '*':
                total = num1 * num2;
                break;
            case '/':
                total = num1 / num2;
                break;
            case '%':
                total = num1 % num2;
                break;
        }
        printf("%d %c %d = %d \n", num1, c, num2, total); // 화면에 출력

    }

    fclose(fr);

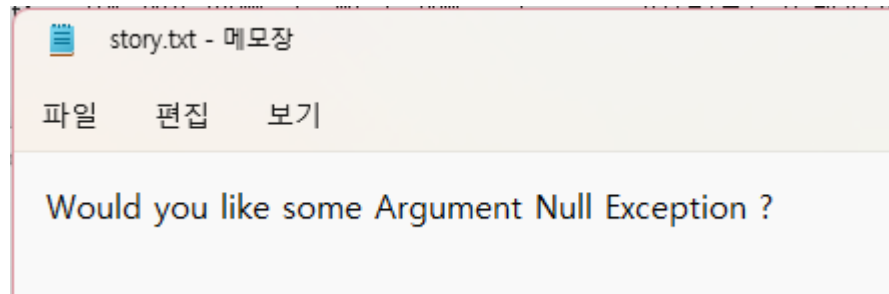
    return 0;
}
```

## 실습 9

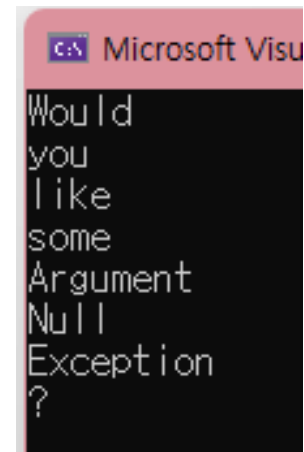
문제 9)

파일에서 문장을 입력받아 단어 단위로 개행하며 출력하시오.

(입력)



(출력)



# 실습 9 정답

```
7 Users > jch99 > Desktop > work > 영문대교강의원 > 교습C프로그래밍 > 강의교안 > 9주차 > 파일입출력2 >
1  /*
2  *      [문제 9]
3  *      파일에서 문장을 입력받아 단어 단위로 개행하며 출력하시오.
4  *
5  *      (파일 입력)
6  *      This file is test file
7  *
8  *      (화면 출력)
9  *      This
10 *      file
11 *      is
12 *      test
13 *      file
14 *
15 */
16
17
18
19 #define _CRT_SECURE_NO_WARNINGS
20 #include <stdio.h>
21 #include <string.h>
22
```



# 실습 9 정답

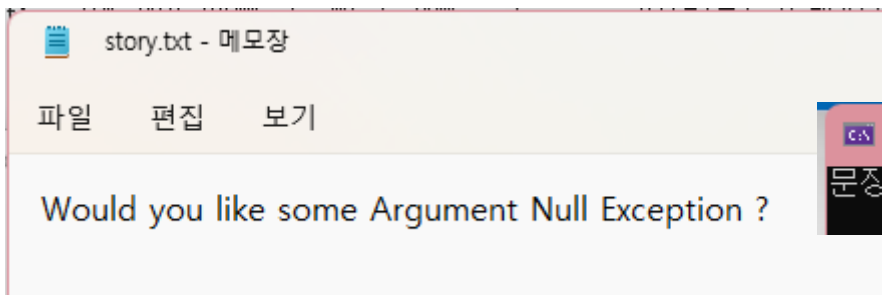
```
18
19 #define _CRT_SECURE_NO_WARNINGS
20 #include <stdio.h>
21 #include <string.h>
22
23 int main() {
24     // 스트림 형성
25     FILE* fw = fopen("story.txt", "r");
26     char str[100];
27
28     while (fscanf(fw, "%s", str) != EOF) { // 파일의 문장이 끝날 때 까지 반복
29
30         char* ptr = strtok(str, " "); // 공백 문자를 기준으로 문자열을 자르고 포인터를 반환
31
32         while (ptr != NULL) { // 자른 문자열이 나오지 않을 때 까지 반복
33
34             printf("%s\n", ptr);
35             ptr = strtok(NULL, " "); // 다음 문자열을 잘라서 포인터를 반환
36
37         }
38     }
39
40 }
41
42 return 0;
43 }
```

## 실습 10

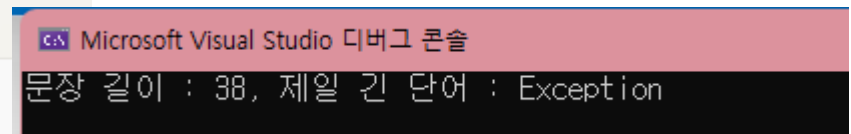
문제 10)

파일에서 문장을 입력받아 문장의 길이를 출력하고(공백 제외),  
제일 긴 단어를 출력하시오.

(입력)



(출력)



# 실습 10 정답

```
1  /*
2  *      [문제 9]
3  *      파일에서 문장을 입력받아 문장의 길이를 출력하고(공백 제외), 제일 긴 단어를 출력하시오.
4  *
5  *      (파일 입력)
6  *      Would you like some Argument Null Exception ?
7  *
8  *      (화면 출력)
9  *      문장 길이 : 38, 제일 긴 단어 : Exception
10 *
11 */
12
13
14 #define _CRT_SECURE_NO_WARNINGS
15 #include <stdio.h>
16 #include <string.h>
```

# 실습 10 정답

```
13
14 #define _CRT_SECURE_NO_WARNINGS
15 #include <stdio.h>
16 #include <string.h>
17
18 int main() {
19
20     FILE* fw = fopen("story.txt", "r");    // 스트림 형성
21
22     char str[100];
23     int max = 0;
24     int total = 0;
25     char max_str[100] = {0};
26
27     while (fscanf(fw, "%s", str) != EOF) { // 파일의 문장이 끝날 때 까지 반복
28
29         char* ptr = strtok(str, " ");    // 공백 문자를 기준으로 문자열을 자르고 포인터를 반환
30
31         while (ptr != NULL) {            // 작은 문자열에 나오지 않을 때 까지 반복
32             if (strlen(ptr) >= max) {    // 같은 단어가 제일 긴 단어일 경우
33                 strcpy(max_str, ptr);
34                 max = strlen(ptr);
35             }
36
37             total += strlen(ptr);        // 문장 길이에 단어 문자 수를 더함
38             ptr = strtok(NULL, " ");    // 다음 문자열을 잘라서 포인터를 반환
39         }
40
41     }
42
43     printf("문장 길이 : %d, 제일 긴 단어 : %s \n", total, max_str);
44
45     return 0;
46 }
```

# 실습 11

문제 11)

fgetc, fputc 함수를 이용해서 getchar, putchar 함수와 동일한 기능의 함수를 정의하시오.

단, 함수의 이름은 getchar\_f, putchar\_f로 정의하고, 함수의 정의가 올바른지 확인하기 위한 적절한 형태의 main 함수도 정의한다.

```
Microsoft Visual Studio 디버그 콘솔
아래는 getchar_f, putchar_f의 test입니다.
z
z

아래는 getchar, putchar의 test입니다.
t
t
```

# 실습 11 정답

```
1  /*
2  *   [문제 11]
3  *   fgetc, fputc 함수를 이용해서 getchar, putchar 함수와 동일한 기능의 함수를 정의하시오.
4  *   단, 함수의 이름은 getchar_f, putchar_f로 정의하고, 함수의 정의가 올바른지 확인하기 위한 적절한 형태의 main 함수도 정의한다.
5  */
6
7
8  #define _CRT_SECURE_NO_WARNINGS
9  #include <stdio.h>
10
11 char getchar_f() {
12     return fgetc(stdin);    // 표준 입력 스트림 사용
13 }
14
15 void putchar_f(char c) {
16     fputc(c, stdout);       // 표준 출력 스트림 사용
17     return;
18 }
19
```

# 실습 11 정답

```
7
8 #define _CRT_SECURE_NO_WARNINGS
9 #include <stdio.h>
10
11 char getchar_f() {
12     return fgetc(stdin);    // 표준 입력 스트림 사용
13 }
14
15 void putchar_f(char c) {
16     fputc(c, stdout);    // 표준 출력 스트림 사용
17     return;
18 }
19
20 int main() {
21
22     // test code.
23     printf("아래는 getchar_f, putchar_f의 test입니다.\n");
24     char test = getchar_f();
25     putchar_f(test);
26
27     getchar_f();    // 버퍼 비움
28
29     printf("\n\n아래는 getchar, putchar의 test입니다.\n");
30
31     int a = getchar();
32     putchar(a);
33
34     printf("\n");
35     return 0;
36 }
```

## 실습 12

### 문제 12)

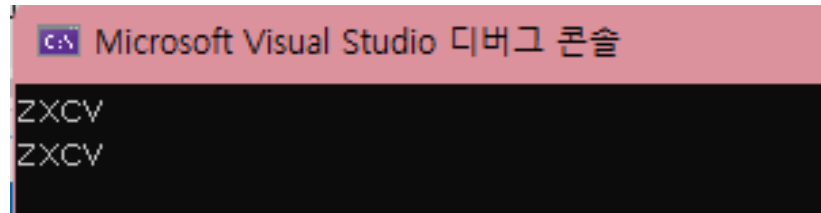
fgetc, fputc를 이용해서 gets함수, puts함수와 동일한 기능의 함수를 정의하시오.  
또, 이를 검증하기 위한 적절한 main 함수를 제작하여 테스트하시오.

아래 규칙을 따라 제작한다.

- 최대 문자 입력은 NULL 문자를 포함하여 64자로 한다.
- gets\_f의 반환 값은 입력받은 버퍼의 주소이다.
- puts\_f의 반환 값은 성공하면 0을 반환하도록 한다. 단, EOF 반환(-1)은 구현하지 않아도 된다.

함수는 아래와 같이 선언한다.

```
char* gets_f(char *buffer);  
int puts_f(const char *string);
```



```
Microsoft Visual Studio 디버그 콘솔  
ZXCV  
ZXCV
```



# 실습 12 정답

```
1  /*
2  *   [문제 12]
3  *   fgetc, fputc를 이용해서 gets 함수, puts 함수와 동일한 기능의 함수를 정의하시오.
4  *   또, 이를 검증하기 위한 적절한 main 함수를 제작하여 테스트하시오.
5  *
6  *   아래 규칙을 따라 제작한다.
7  *   - 최대 문자 입력은 NULL 문자를 포함하여 64자로 한다.
8  *   - gets_f의 반환 값은 입력받은 버퍼의 주소이다.
9  *   - puts_f의 반환 값은 성공하면 0을 반환하도록 한다. 단, EOF 반환(-1)은 구현하지 않아도 된다.
10 *
11 *   함수는 아래와 같이 선언한다.
12 *   char* gets_f(char *buffer);
13 *   int puts_f(const char *string);
14 *
15 */
```

# 실습 12 정답

```
17  #define _CRT_SECURE_NO_WARNINGS
18
19  #include <stdio.h>
20  #define MAX_LENGTH 64      // 최대 입력 제한
21
22  char* gets_f(char* buffer) {
23      int i = 0;
24
25      // 표준 입력 스트림에서 MAX_LENGTH만큼 문자를 저장
26      for (i = 0; i < MAX_LENGTH - 1; i++) {
27          buffer[i] = fgetc(stdin);      // 한 글자 입력
28          if (buffer[i] == '\n') break;  // 개행 문자 발견시 종료
29      }
30      buffer[i] = 0;                    // 맨 마지막 NULL 문자 추가
31
32      return buffer;                    // 버퍼의 주소 다시 반환
33  }
34
35  int puts_f(const char* string) {
36
37      // 표준 출력 스트림으로 입력받은 string을 MAX_LENGTH만큼 출력
38      for (int i = 0; i < MAX_LENGTH - 1; i++) {
39          fputc(string[i], stdout);      // 한 글자 출력
40          if (string[i] == 0) break;     // NULL 문자 발견시 종료
41      }
42
43      printf("\n");                      // 개행
44      return 0;
45  }
```

# 실습 12 정답

```
40
47
48 int main() {
49     char str[MAX_LENGTH];
50
51     // test code.
52     gets_f(str);
53     puts_f(str);
54
55     return 0;
56 }
```

## 실습 13

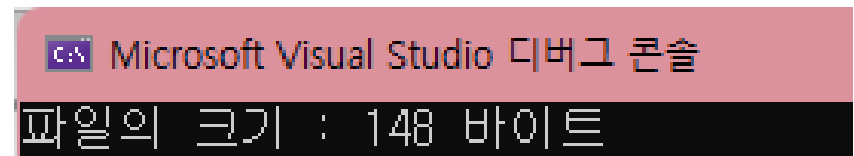
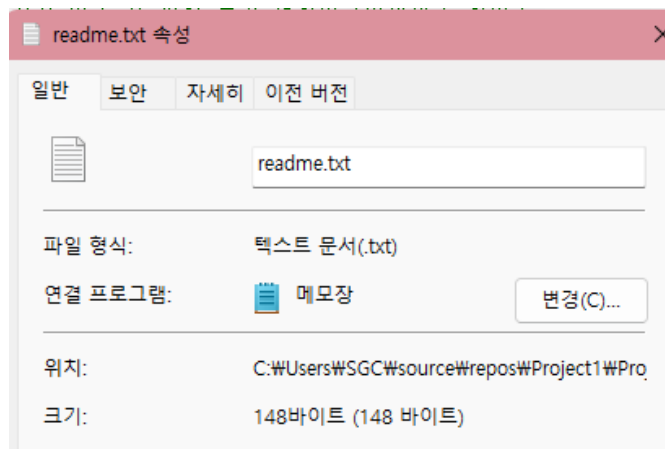
### 문제 13)

파일의 크기를 바이트 단위로 계산하여 반환하는 함수를 제작하시오.

해당 함수는 아래와 같이 정의한다.

```
long GetFileSize(FILE* fp);
```

단, `fseek` 함수를 이용하여 `GetFileSize` 함수를 구현해야 하며,  
함수가 호출된 이후에도 파일 위치 지시자의 정보는 변경되면 안 된다.  
(파일의 크기를 계산하는 과정에서 변경시켰다면, 파일의 크기를 계산한 이후에  
되돌려 놓아야 한다.)



# 실습 13 정답

```
1  /*
2  *   [문제 13]
3  *   파일의 크기를 바이트 단위로 계산하여 반환하는 함수를 제작하시오.
4  *   해당 함수는 아래와 같이 정의한다.
5  *   long GetFileSize(FILE* fp);
6  *
7  *   단, ftell 함수를 이용하여 GetFileSize 함수를 구현해야 하며,
8  *   함수가 호출된 이후에도 파일 위치 지시자의 정보는 변경되면 안 된다.
9  *   (파일의 크기를 계산하는 과정에서 변경시켰다면, 파일의 크기를 계산한 이후에 되돌려 놓아야 한다.)
10 *
11 *
12 */
13
14 #define _CRT_SECURE_NO_WARNINGS
15 #include <stdio.h>
16 #include <string.h>
17
18 long GetFileSize(FILE* fr) {
19     long fpos;
20     long fsize;
21
22     fpos = ftell(fr);           // 파일 위치 지시자의 위치 정보 백업
23
24     fseek(fr, 0, SEEK_END);     // 파일 위치 지시자를 파일 맨 끝으로 옮김
25     fsize = ftell(fr);         // 파일 위치 지시자의 위치 정보 저장
26
27     fseek(fr, fpos, SEEK_SET);  // 파일 위치 지시자를 원위치로 복구시킴
28
29     return fsize;
30 }
```

# 실습 13 정답

```
31
32 int main() {
33
34     char str[100];
35
36     FILE* fr = fopen("readme.txt", "r");
37
38     // 파일 크기 확인
39     printf("파일의 크기 : %ld 바이트\n", GetFileSize(fr));
40
41     // 파일 위치 지시자 이동 후 파일 크기 재확인 (정상작동 확인)
42     fgets(str, 100, fr);
43     fputs(str, stdout);
44     printf("파일의 크기 : %ld 바이트\n", GetFileSize(fr));
45
46     // 파일 위치 지시자 이동 후 파일 크기 재확인2 (정상작동 확인)
47     fgets(str, 100, fr);
48     fputs(str, stdout);
49     printf("파일의 크기 : %ld 바이트\n", GetFileSize(fr));
50
51     return 0;
52 }
```

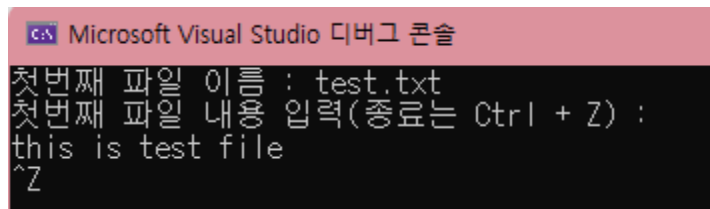
## 실습 14

문제 14)

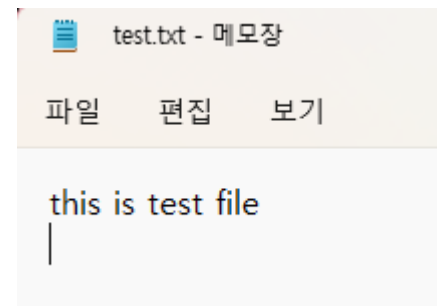
파일 이름을 입력받아 해당 파일 이름으로 파일을 생성하고,  
내용을 입력받는 프로그램을 제작하시오.

(힌트)

콘솔 창에서 `getchar` 함수 사용 도중에 `Ctrl + Z` 를 통해 EOF 값을  
발생 시킬 수 있음.



```
Microsoft Visual Studio 디버그 콘솔
첫번째 파일 이름 : test.txt
첫번째 파일 내용 입력(종료는 Ctrl + Z) :
this is test file
^Z
```



```
test.txt - 메모장
파일 편집 보기
this is test file
|
```

# 실습 14 정답

[문제 15]

두 파일을 생성하여 각 파일에 문자들을 입력받아 저장하고, 두 파일의 내용이 동일한지 판별하는 프로그램을 작성하시오.  
단, 문제 14에서 제작한 '파일 이름을 입력받아 해당 파일 이름으로 파일을 생성하는 프로그램'도 사용하여 제작한다.

(입력 예시)

>> 첫번째 파일 이름 : 1.txt

>> 첫번째 파일 내용 입력(종료는 Ctrl + Z) : asdf

>> ^Z

>> 두번째 파일 이름 : 2.txt

>> 두번째 파일 내용 입력(종료는 Ctrl + Z) : asdfzxcv

>> ^Z

(출력 예시, 개행 문자는 공백으로 처리한다)

1 : a a

2 : s s

3 : d d

4 : f f

5 : z

5번째에서 , z 다름

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

```
#define FILENAME_LEN 50
```



# 실습 14 정답

```
28 int main(void) {
29
30     // fp1, fp2 : 파일 포인터
31     // fname1, fname2 : 파일 이름을 지정할 위한 변수
32     // ch : 입력을 위한 변수
33     // ch1, ch2 : 문장 체크를 위한 변수
34
35     FILE* fp1 = NULL, * fp2 = NULL;
36     char fname1[FILENAME_LEN] = {0};
37     char fname2[FILENAME_LEN] = {0};
38     char ch, ch1, ch2;
39
40     // 파일 생성, 데이터 입력
41     printf("첫번째 파일 이름 : ");
42     gets(fname1, FILENAME_LEN);
43     fopen_s(&fp1, fname1, "w");
44     printf("첫번째 파일 내용 입력(종료는 Ctrl + Z) : \n");
45     while ((ch = getchar()) != EOF)
46         fputc(ch, fp1);
47
48     printf("두번째 파일 이름 : ");
49     gets(fname2, FILENAME_LEN);
50     fopen_s(&fp2, fname2, "w");
51     printf("두번째 파일 내용 입력(종료는 Ctrl + Z) : \n");
52     while ((ch = getchar()) != EOF)
53         fputc(ch, fp2);
54
55     // 스트림 소멸
56     fclose(fp1);
57     fclose(fp2);
58 }
```

// 파일 이름 입력  
// 입력받은 파일 이름으로 write 파일 스트림 생성

// getchar가 EOF(Ctrl + Z)를 입력받으면 종료  
// 입력받은 문자를 파일로 출력

# 실습 14 정답

```
59 // 생성된 파일을 다시 열어 내용이 동일한지 비교
60 fopen_s(&fp1, fname1, "r"); // fname1의 read 파일 스트림 생성
61 fopen_s(&fp2, fname2, "r"); // fname2의 read 파일 스트림 생성
62 if (fp1 == NULL || fp2 == NULL) { // 파일을 열지 못했다면
63     fprintf(stderr, "파일 열기 실패\n");
64     return -1;
65 }
66
67 int n = 0;
68 while (!feof(fp1)) { // fname1 파일의 끝 기준
69     n++; // 몇 번째 글자인지 확인을 위한 변수
70
71     // 각 스트림에서 입력받은 글자 비교
72     ch1 = fgetc(fp1);
73     ch2 = fgetc(fp2);
74
75     if (ch1 == EOF || ch2 == EOF) // 내용이 같고, 끝났을 경우 종료
76         break;
77
78     if (ch1 == '\n') ch1 = ' '; // 개행 문자 예외 처리
79     if (ch2 == '\n') ch2 = ' ';
80
81     printf("%d : %c %c \n", n, ch1, ch2);
82     if (ch1 != ch2) { // 글자가 다르다면 종료
83         printf("%d번째에서 %c, %c 다름\n", n, ch1, ch2);
84         break;
85     }
86 }
87
88
89 // 스트림 소멸
90 fclose(fp1);
91 fclose(fp2);
92
93 return 0;
94 }
```

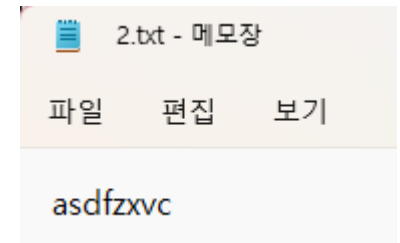
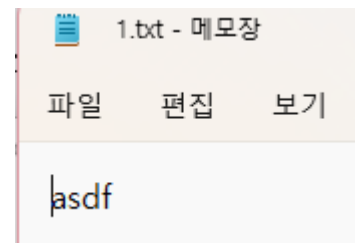
# 실습 15

## 문제 15)

두 파일을 생성하여 각 파일에 문장들을 입력받아 저장하고,  
두 파일의 내용이 동일한지 파일에서 다시 읽어와 판별하는  
프로그램을 작성하시오.

단, 문제 14에서 제작한 '파일 이름을 입력받아 해당 파일 이름으로  
파일을 생성하는 프로그램'도 사용하여 제작한다.

```
Microsoft Visual Studio 디버그 콘솔
첫번째 파일 이름 : 1.txt
첫번째 파일 내용 입력(종료는 Ctrl + Z) :
asdf
^Z
두번째 파일 이름 : 2.txt
두번째 파일 내용 입력(종료는 Ctrl + Z) :
asdfzxvc
^Z
1 : a a
2 : s s
3 : d d
4 : f f
5 : z
5번째에서 , z 다음
```



# 실습 15 정답

```
1  /*
2  *   [문제 15]
3  *   두 파일을 생성하여 각 파일에 문자들을 입력받아 저장하고, 두 파일의 내용이 동일한지 판별하는 프로그램을 작성하시오.
4  *   단, 문제 14에서 제작한 '파일 이름을 입력받아 해당 파일 이름으로 파일을 생성하는 프로그램'도 사용하여 제작한다.
5  *
6  *   (입력 예시)
7  *   >> 첫번째 파일 이름 : 1.txt
8  *   >> 첫번째 파일 내용 입력(종료는 Ctrl + Z) : asdf
9  *   >> ^Z
10 *   >> 두번째 파일 이름 : 2.txt
11 *   >> 두번째 파일 내용 입력(종료는 Ctrl + Z) : asdfzxcv
12 *   >> ^Z
13 *
14 *   (출력 예시, 개행 문자는 공백으로 처리한다)
15 *   1 : a a
16 *   2 : s s
17 *   3 : d d
18 *   4 : f f
19 *   5 :   z
20 *   5번째에서 , z 다음
21 *
22 */
23
24
25 #include <stdio.h>
26 #define FILENAME_LEN 50
```

# 실습 15 정답

```
28 int main(void) {
29
30     // fp1, fp2 : 파일 포인터
31     // fname1, fname2 : 파일 이름 지정을 위한 변수
32     // ch : 입력을 위한 변수
33     // ch1, ch2 : 문장 체크를 위한 변수
34
35     FILE* fp1 = NULL, * fp2 = NULL;
36     char fname1[FILENAME_LEN] = {0};
37     char fname2[FILENAME_LEN] = {0};
38     char ch, ch1, ch2;
39
40     // 파일 생성, 데이터 입력
41     printf("첫번째 파일 이름 : ");
42     gets(fname1, FILENAME_LEN);
43     fopen_s(&fp1, fname1, "w");
44     printf("첫번째 파일 내용 입력(종료는 ctrl + z) : \n");
45     while ((ch = getchar()) != EOF)
46         fputc(ch, fp1);
47
48     printf("두번째 파일 이름 : ");
49     gets(fname2, FILENAME_LEN);
50     fopen_s(&fp2, fname2, "w");
51     printf("두번째 파일 내용 입력(종료는 ctrl + z) : \n");
52     while ((ch = getchar()) != EOF)
53         fputc(ch, fp2);
54
55     // 스트림 소멸
56     fclose(fp1);
57     fclose(fp2);
58 }
```

// 파일 이름 입력  
// 입력받은 파일 이름으로 write 파일 스트림 생성

// getchar가 EOF(ctrl + z)를 입력받으면 종료  
// 입력받은 문자를 파일로 출력

# 실습 15 정답

```
// 생성된 파일을 다시 열어 내용이 동일한지 비교
```

```
fopen_s(&fp1, fname1, "r");
```

```
fopen_s(&fp2, fname2, "r");
```

```
if (fp1 == NULL || fp2 == NULL) {  
    fprintf(stderr, "파일 열기 실패\n");  
    return -1;  
}
```

```
int n = 0;
```

```
while (!feof(fp1)) {  
    n++;
```

```
    // 각 스트림에서 입력받은 글자 비교
```

```
    ch1 = fgetc(fp1);
```

```
    ch2 = fgetc(fp2);
```

```
if (ch1 == EOF || ch2 == EOF)  
    break;
```

```
if (ch1 == '\n')    ch1 = ' ';
```

```
if (ch2 == '\n')    ch2 = ' ';
```

```
printf("%d : %c %c \n", n, ch1, ch2);
```

```
if (ch1 != ch2) {
```

```
    printf("%d번째에서 %c, %c 다름\n", n, ch1, ch2);
```

```
    break;
```

```
}
```

```
}
```

```
// 스트림 소멸
```

```
fclose(fp1);
```

```
fclose(fp2);
```

```
// fname1의 read 파일 스트림 생성  
// fname2의 read 파일 스트림 생성  
// 파일을 열지 못했다면
```

```
// fname1 파일의 줄 기준  
// 몇 번째 글자인지 확인을 위한 변수
```

```
// 내용이 같고, 끝났을 경우 종료
```

```
// 개행 문자 예외 처리
```

```
// 글자가 다르다면 종료
```

# 실습 15 정답

```
59 // 생성된 파일을 다시 열어 내용이 동일한지 비교
```

```
60 fopen_s(&fp1, fname1, "r");
```

```
62 fopen_s(&fp2, fname2, "r");
```

```
63 if (fp1 == NULL || fp2 == NULL) {
64     fprintf(stderr, "파일 열기 실패\n");
65     return -1;
66 }
```

```
67
68 int n = 0;
```

```
69 while (!feof(fp1)) {
70     n++;
```

```
71 // 각 스트림에서 입력받은 문자 비교
```

```
72 ch1 = fgetc(fp1);
```

```
74 ch2 = fgetc(fp2);
```

```
75
76 if (ch1 == EOF || ch2 == EOF)
77     break;
```

```
78
79 if (ch1 == '\n')    ch1 = ' ';
```

```
80 if (ch2 == '\n')    ch2 = ' ';
```

```
81
82 printf("%d : %c %c \n", n, ch1, ch2);
```

```
83 if (ch1 != ch2) {
84     printf("%d번째에서 %c, %c 다름\n", n, ch1, ch2);
85     break;
86 }
```

```
87
88 }
```

```
// fname1의 read 파일 스트림 생성
// fname2의 read 파일 스트림 생성
// 파일을 열지 못했다면
```

```
// fname1 파일의 줄 기준
// 몇 번째 글자인지 확인을 위한 변수
```

```
// 내용이 같고, 끝났을 경우 종료
```

```
// 개행 문자 예외 처리
```

```
// 글자가 다르다면 종료
```

```
88 // 스트림 소멸
```

```
89 fclose(fp1);
```

```
90 fclose(fp2);
```

```
91
92 return 0;
```

```
93
94 }
```

# 질문 및 정리

