

9 장. 파일 입력과 출력

□ 개념 확인 학습

1. 파일 입출력을 처리하는 과정을 설명해 보세요.
2. `fopen()` 함수의 사용법을 설명해 보세요.
3. `fopen()` 함수에서 파일을 여는 방법에는 어떤 것이 있나요?
4. `fclose()` 함수의 역할은 무엇인가요?
5. 파일 저장 시 텍스트 모드로 처리했을 때와 바이너리 모드로 처리했을 때의 차이점은 무엇인가요?
6. 파일 입출력 시 형식을 지정하여 입출력하는 함수에는 어떤 것이 있나요?
7. `fread()`와 `fwrite()` 함수에서 매개 변수들에 사용되는 파라미터들은 어떤 것이 있으며 사용법은 어떻게 되나요?

8. 어떻게 만들어진 파일이든 파일의 끝에 저장되는 값은 무엇인가요?
9. 파일의 끝인지 아닌지를 판별해 주는 함수는 무엇인가요?
10. feof() 함수를 사용할 때 주의할 점은 무엇인가요?
11. 파일 포인터의 위치를 랜덤하게 변경시키고자 할 때 사용하는 함수는 무엇인가요?
12. 파일 포인터의 위치를 랜덤하게 변경시키고자 할 때 기준점을 표현하는 방법에는 어떤 것들이 있나요?
13. 파일의 크기를 알아내는 방법을 설명 해 보세요.

☐ 적용 확인 학습

1. 다음 서술 내용이 바르면 O, 그렇지 않으면 X를 표시 하세요.

- A. 파일 포인터는 FILE 구조체 타입의 포인터이다. ()
- B. 파일은 텍스트(text) 타입 으로도 이진(binary) 타입 으로도 저장될 수 있다. ()
- C. 상수 stdin은 표준 입력, stdout은 표준 출력을 의미한다. ()
- D. 파일 처리가 끝나면 반드시 fclose() 함수를 이용해 파일 스트림을 닫아 주어야 한다. ()
- E. 파일 모드 중 ‘a’ 를 사용한 경우 파일이 존재하지 않으면 오류가 발생한다. ()
- F. 파일 모드 중 ‘a’ 를 사용하면 파일의 마지막에 내용을 추가한다. ()
- G. 파일 모드 중 ‘rb’ 를 사용하면 파일에서 바이너리 타입으로 내용을 읽어 들이며, 만약 파일이 존재하지 않으면 오류가 발생한다. ()
- H. fprintf(stdin, ...)과 printf(...)은 같은 의미로 사용될 수 있다. ()
- I. 이진(binary) 모드로 저장된 파일을 메모장으로 열면 저장하려던 내용과 같은 값이 들어가 있음을 명시적으로 확인 할 수 있다. ()
- J. fprintf(stdout, “%d” , 정수형 변수)로 저장된 파일을 메모장으로 열면 시스템 내부에서 사용하는 이상한 모양이 들어가 있어 해당 정수 값을 눈으로 확인 할 수 없다. ()
- K. fwrite() 함수와 fread() 함수는 자료의 입출력을 텍스트 모드로 처리한다. ()

L. fwrite() 함수와 fread() 함수의 첫 번째 매개변수는 자료가 저장될 변수의 주소이다. ()

M. fseek() 함수에서 기준점으로 사용하는 상수인 SEEK_SET, SEEK_CUR, SEEK_END는 정수형 기호상수로 각각 1, 2, 3 이다. ()

N. ftell() 함수는 현재 파일 포인터의 위치를 바이트 단위로 알려준다. ()

2. 다음에 서술된 문장에 어울리는 프로그램 코드를 완성해 보세요.

A. 파일 포인터 fp와 연결된 파일에 텍스트 모드로 num의 값을 쓴다.

```
int num=5;
```

B. 파일 포인터 fp와 연결된 파일에서 바이너리 모드로 한 개의 정수형 자료를 num 변수로 읽어 들인다.

```
int num;
```

C. 다섯 개의 정수형 자료를 저장하고 있는 num 변수의 내용을 파일 포인터 fp와 연결된 파일에 바이너리 모드로 쓴다.

```
int num[5] = { 5, 4, 3, 2, 1 };
```

- D. 파일 포인터 fp와 연결된 파일에서 바이너리 모드로 다섯 개의 정수형 자료를 num 변수로 읽어 들인다.

```
int num[5];
```

- E. 파일 포인터 fp와 연결된 파일에 정수가 여러 개 저장되어 있고, 파일 포인터가 파일 끝에 위치 할 때, 파일의 첫 번째 정수를 읽어 num에 저장한다.

```
int num;
```

- F. 파일 포인터 fp와 연결된 파일에 정수가 여러 개 저장되어 있을 때 가장 마지막에 저장된 정수를 읽어 드린다.

```
int num;
```

- G. 파일의 크기를 size 변수에 byte 단위로 저장한다.

```
int size;
```

- H. person 구조체가 다음과 같을 때, student 변수의 내용을 파일 포인터 fp와 연결된 파일에 쓴다.

```
struct person {
    char name[20];
    int yyyymmdd;
};
struct person student = { "daniel" , 20201010 };
```

- I. 위 H번에 저장된 내용을 읽어 friend 변수에 저장한다.

```
struct person friend;
```

☐ 응용 프로그래밍

1. 아래의 실행 예처럼 키보드로 부터 'q'를 입력 받기 전까지 계속 문자를 입력 받아, 입력 받은 문자들을 파일로 출력하는 프로그램을 작성하세요. 단, 'q'는 파일에 저장되지 않습니다. 저장 할 파일의 이름은 chars.txt로 합니다. (힌트 : fputc(), fprintf(), fwrite() 사용 가능.)

문자 입력('q' 이면 종료) : abcdefghijkq
chars.txt 파일에 저장했습니다.



2. 아래의 실행 예처럼 위 1번에서 저장된 chars.txt 파일의 내용을 읽어 화면으로 출력하는 프로그램을 작성하세요. 프로그램 작성 방법에 따라 마지막 ‘k’ 문자가 두 번 출력 될 수 있으니 반드시 한 번만 출력되는지 확인하세요. (힌트 : fgetc(), fscanf(), fread() 사용 가능.)

```
chars.txt 파일을 읽습니다.  
abcdefghijkl
```

3. 키보드로 부터 <enter>만을 입력 받기 전까지 계속 문자열을 입력 받아, 입력 받은 문자열들을 파일로 출력하는 프로그램을 작성하세요. 단, 저장할 파일의 이름은 strings.txt로 합니다. (힌트 : fputs(), fprintf(), fwrite() 사용 가능.)

```
문자열 입력 : apple  
문자열 입력 : candy  
문자열 입력 : vitamin  
문자열 입력 :  
입력하신 문자열을 strings.txt로 저장했습니다.
```

4. 위 3번에서 저장된 strings.txt 파일의 내용을 읽어 화면으로 출력하는 프로그램을 작성하세요. (힌트 : fgetc(), fscanf(), fread() 사용 가능.)

```
strings.txt를 읽습니다.  
apple  
candy  
vitamin
```

5. 위 1번과 3번에서 저장된 chars.txt와 strings.txt의 내용이 일치하는지 아닌

지를 판별하고 그 결과를 화면으로 출력하는 프로그램을 작성하세요.

chars.txt를 열었습니다. strings.txt를 열었습니다. 다릅니다.	chars.txt를 열었습니다. strings.txt를 열었습니다. 같습니다.
---	---

6. 위 3번의 strings.txt의 내용을 samestrings.txt에 복사 한 후, samestrings.txt의 내용을 읽어 화면에 출력하는 프로그램을 작성세요.

strings.txt를 열었습니다. samestrings.txt를 열었습니다. strings.txt의 내용을 samestrings.txt에 복사했습니다. samestrings.txt를 읽습니다. apple candy vitamin
--

7. double형 자료 다섯 개를 저장하는 배열을 생성한 후 그 배열의 내용을 doubles.data 파일에 출력하는 프로그램을 작성하세요.

doubles.data를 열었습니다. 1.100000 2.200000 3.300000 4.400000 5.500000 double형 배열의 내용을 배열로 출력했습니다.

8. 위 7번에서 생성된 doubles.data 파일에서 마지막에 저장된 double 값을 읽어 화면에 출력하는 프로그램을 작성세요.

doubles.data를 열었습니다.

1.1 2.2 3.3 4.4 5.5

double형 배열의 내용을 출력했습니다.

ftell = 40

doubles.data의 마지막 데이터에 접근 합니다.

ftell = 32

마지막에 저장된 double 값 = 5.5

9. 아래의 제시된 main() 함수와 실행 예처럼 데이터를 입력 받아 fruit 구조체에 저장한 후 자료가 저장된 구조체를 fruit.data 파일에 저장하는 프로그램을 작성하세요. 단, 파일 이름인 fruit.data는 main() 함수의 매개변수에 저장하여 사용합니다.

//응용 프로그래밍 9번 10번 관련

```
int main(int argc, char *argv[])
{
    fruit datas[3];
    int len = sizeof(datas)/sizeof(fruit);

    input_datas(datas, len);
    puts("\nAfter input_datas");
    print_datas(datas, len);

    write_file(datas, len, argv[1]);

    read_file(datas, len, argv[1]);
    puts("\nAfter read_file");
    print_datas(datas, len);

    int total = total_amount(datas,
len);
    printf("\n총 판매금액 = %d\n",
total);
}
```



```
C:\lecture_src\ctest>ctest fruit.data
과일종류 : apple
단가 : 111.1
판매량 : 20

과일종류 : banana
단가 : 99.5
판매량 : 33

과일종류 : orange
단가 : 155.5
판매량 : 22

After input_datas
=====
과일   단가   판매량
=====
apple  111.10  20
banana 99.50   33
orange 155.50   22
=====

After read_file
=====
과일   단가   판매량
=====
apple  111.10  20
banana 99.50   33
orange 155.50   22
=====

총 판매금액 = 8925
C:\lecture_src\ctest>
```

== 해법 : 함수화 전 ==

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

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

```
#include <time.h>
```

```
struct fruit {
```

```
    char name[20]; //과일이름 name(문자열),
```

```
    double price; //단가
```

```
    int sales; //판매량
```

```
};
```

```
typedef struct fruit fruit;
```

```
void ex_09(int argc, char *argv[])
```

```
{
```

```
    fruit wdatas[3], rdatas[3], *p;
```

```
    FILE *fp;
```

```
    int len = sizeof(wdatas)/sizeof(fruit);
```

```
    p=wdatas;
```

```
    puts(">>Before input_datas");
```

```
    for(int i=0; i<len; i++) {
```

```
        printf("과일종류 : ");
```

```
        gets((p+i)->name);
```

```
        printf("단가 : ");
```

```
        scanf("%lf", &(p+i)->price);
```

```
        printf("판매량 : ");
```

```
        scanf("%d", &(p+i)->sales);
```

```
        while (getchar() != '\n'); //버퍼에 저장되어 있는 값 삭제
```

```
        puts("");
```

```
    }
```

```
    puts(">>After input_datas");
```

```
    printf("=====\\n");
```

```
    printf("과일\t단가\t판매량\\n");
```

```
    printf("=====\\n");
```

```

for(int i=0; i<len; i++) {
    printf("%s\t%.2lf\t%d\n", (p+i)->name, (p+i)->price, (p+i)->sales);
}
printf("=====\n");

puts(">>Before write_file");
fp = fopen(argv[1], "wb");
fwrite(wdatas, sizeof(wdatas), 1, fp);
fclose(fp);
puts(">>After write_file");

puts(">>Before read_file");
fp = fopen(argv[1], "rb");
fread(rdatas, sizeof(rdatas), 1, fp);
fclose(fp);

puts(">>After read_file");
p = rdatas;
printf("=====\n");
printf("과일\t단가\t판매량\n");
printf("=====\n");
for(int i=0; i<len; i++) {
    printf("%s\t%.2lf\t%d\n", (p+i)->name, (p+i)->price, (p+i)->sales);
}
printf("=====\n");

puts(">>Before 총 판매금액");
int total = 0;
p = rdatas;
for(int i=0; i<len; i++) {
    total += (p+i)->price * (p+i)->sales;
}
printf("총 판매금액 = %d\n", total);
}

int main(int argc, char *argv[]) {
    ex_09(argc, argv);
    return 0;
}

```

10. 위에 제시된 main() 함수와 실행 예처럼 fruit.data 파일에 저장된 내용을 읽어 매출 총액을 화면에 출력하는 프로그램을 작성세요.