PART 14 파일 입출력 연습문제 풀이

1. 다음 설명으로 옳지 않은 것은? ④

① fopen 함수로 파일 열기를 하면 FILE 포인터가 반환된다.
② fgetc 함수를 이용하여 파일로부터 한 문자씩 읽을 수 있다.
③ fgets 함수를 이용하여 파일로부터 한 줄씩 읽을 수 있다.
④ fopen 함수로 파일을 열 때 오류가 발생하면 -1를 반환한다.

2. 다음 중 파일로부터 한 줄씩 읽기 위해 사용할 수 있는 함수는? ③

① fopen
② fgetc
③ fgets
④ fscanf

3. 다음 중 fopen의 텍스트 파일 입출력 모드에 대한 설명으로 옳지 않은 것은? ③

① "r"은 읽기 전용모드로 해당 파일이 없으면 NULL을 반환한다.
② "r+"는 읽기와 쓰기 모드로 해당 파일이 없으면 NULL을 반환한다.
③ "w+"는 읽기와 쓰기 모드로 해당 파일이 없으면 NULL을 반환한다.
④ "w"는 쓰기 전용모드로 해당 파일이 없으면 파일을 생성한다.

4. 다음 괄호 안에 알맞은 말을 넣으시오.

① C 파일은 모든 데이터를 연속된 (바이트) 형태로 저장한다.
② 파일을 사용하기 위해서는 반드시 (파일 열기(fopen))(을)를 먼저 해야 한다.
③ 파일 열기를 하면 (FILE 포인터)(이)가 반환된다.
④ 열린 파일에서 다음 읽거나 기록할 파일 내 위치를 (파일 위치)(이)라고 하며 시스템 내에 (FILE 구조체)(이)가 그 파일의 파일 위치를 저장하고 있다.

5. 다음 표는 fopen의 텍스트 파일 입출력 모드에 대한 설명이다. 빈칸을 채우시오.

모드	의미	파일이 없으면	파일이 있으면
"r"	읽기 전용(read)	NULL 반환	정상 동작
"w"	쓰기 전용(write)	새로 생성	기존 내용 삭제
"a"	추가 쓰기(append)	새로 생성	기존 내용 뒤에 추가
"r+"	읽기와 쓰기	NULL 반환	정상 동작
"w+"	읽기와 쓰기	새로 생성	기존 내용 삭제
"a+"	추가를 위한 읽기와 쓰기	새로 생성	기존 내용 뒤에 추가

6. 다음 표는 파일 입출력 함수의 기능에 대한 설명이다. 빈칸을 채우시오.

파일 입출력 함수	기능
fgetc(), getc()	문자 단위로 입력하는 함수
fputc(), putc()	문자 단위로 출력하는 함수
fgets()	줄 단위로 입력하는 함수
fputs()	줄 단위로 출력하는 함수
fscanf()	포맷에 따라 자료를 입력하는 함수
fprintf()	포맷에 따라 자료를 출력하는 함수

7. 다음 3개의 파일 포인터에 대해서 설명하시오.

① stdin 표준입력에 대한 FILE 포인터, 가리키는 장치는 키보드이다.
② stdout 표준출력에 대한 FILE 포인터, 가리키는 장치는 모니터이다.
③ stderr 표준오류에 대한 FILE 포인터, 가리키는 장치는 모니터이다.

8. 파일 복사 프로그램을 참고로 하여 파일 추가 프로그램 append를 작성하여라. 이 프로그램의 사용법은 다음과 같으며 file1의 내용을 file2의 내용 뒤에 덧붙인다.

```
append file1 file2
```

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *fp1, *fp2;
    int c;
```

```
    fp1 = fopen(argv[1], "r");
    fp2 = fopen(argv[2], "a");

    while ((c = fgetc(fp1)) != EOF)
         fputc(c, fp2);

    return 0;
}
```

9. fseek() 함수를 이용해서 텍스트 파일을 역순으로 출력하는 프로그램을 작성하여라.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *fp;
    int c, i, n;

    fp = fopen(argv[1], "r");

    fseek(fp, 0L, SEEK_END);
    n = ftell(fp);
    for (i = n-1; i >= 0; i--) {
         fseek(fp, i, SEEK_SET);
         c = fgetc(fp);
         putchar(c);
    }
    return 0;
}
```

10. 파일에 있는 문자의 개수, 단어의 개수, 줄의 개수를 계산하여 알려주는 프로그램을 작성하여라.

```
#include <stdio.h>
#define IN 1
#define OUT 0

int main(int argc, char *argv[])
{
    FILE *fp;
    int c, nl, nw, nc, state;

    fp = fopen(argv[1], "r");
    state = OUT;
    nl = nw = nc = 0;
```

```
    while ((c = fgetc(fp)) != EOF) {
         nc++;
         if (c == '\n')
        nl++;
         if (c == ' ' || c == '\n' || c == '\t')
        state = OUT;
         else if (state == OUT) {
        state = IN;
        nw++;
         }
    }
    printf("%d %d %d\n",nl, nw, nc);
    return 0;
}
```

11. 두개의 파일을 비교하는 프로그램을 작성하여라. 파일 시작부터 비교하면서 달라지는 처음 줄을 출력하여라.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *fp1, *fp2;
    int c1,c2;
    char s1[100],s2[100];

    fp1 = fopen(argv[1], "r");
    fp2 = fopen(argv[2], "r");

    while ((c1 = fgetc(fp1)) != EOF && (c2 = fgetc(fp2)) != EOF) {
         if (c1 != c2) {
        fgets(s1, 100, fp1);
        fgets(s2,100, fp2);
        printf("두 파일은 서로 다릅니다.\n");
        printf("파일 %s : %s\n", argv[1], s1);
        printf("파일 %s : %s\n", argv[2], s2);
        break;
         }
    }
    return 0;
}
```

12. [프로그램 14-10]은 학생 레코드가 학번 순으로 저장된 파일에 대해서만 동작하도록 작성되었다. 임의의 학번 순으로 저장된 파일에 대해서도 동작하도록 이 프로그램을 확장하여라.

```c
//*********************************************************************
// newstudent3.c
//
// 파일에 저장된 특정 학생의 정보를 읽어서 출력한다.
//*********************************************************************

#include <stdio.h>
#include "student.h"
#define START_ID 601001

int search(int ar[], int n, int x)
{   int i;
    for (i = 0; i < n; i++)
        if (x == ar[i])
            return i;
    return -1;
}

int main()
{
    struct student st, *stp = &st;
    int stid, pos, nread, ids[100], i, n=0;
    FILE *fp = fopen("st_file", "rb");

    if(fp == NULL ) {
        printf("파일 열기 오류.\n");
        exit(1);
    }

    nread = fread(stp, sizeof(struct student), 1, fp);
    while (nread != 0) {
        ids[n++] = stp->stud_id;
        nread = fread(stp, sizeof(struct student), 1, fp);
    }

    printf("찾을 학생의 학번 입력: “);
    while (scanf("%d", &stid) == 1) {
        i = search(ids, n, stid);     // ids 내에서 stid의 순서 찾기
        pos = i * sizeof(struct student);
        fseek(fp, pos,SEEK_SET);
        fread(stp, sizeof(struct student), 1, fp);
        printf(“학번: %10d 이름: %4s 학년: %4d 전공: %8s\n",
            st.stud_id, st.name, st.year, st.major);
        printf("찾을 학생의 학번 입력: “);
    }
    fclose(fp);
```

```
    return 0;
}
```