```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

#define LINE_LENGTH     255

#define BUFSIZE         1024
#define NAMESIZE        255

#define DISPLAY         1
#define COPY            2
#define APPEND          3
#define QUIT            4

#define PERMS   0644

void menu_out();
int get_filename(char *pathname);
int f_display(const char *pathname);
int f_copy(const char *original, const char *target);
int f_append(const char *original, const char *target);

/*****************************************************************************/
int main()
{
        int end_of_program = 0;
        int choice;
        char line[LINE_LENGTH];
        char filename1[NAMESIZE];
        char filename2[NAMESIZE];

        while(!end_of_program) {
                menu_out();
                printf("Enter your choice : ");
                if (scanf("%d", &choice) != 1) {        /* 잘못된 입력 처리 */
                        scanf("%s", line);
                        continue;
                }

                switch (choice) {
                        case DISPLAY :
                                if (get_filename(filename1)) {
                                        f_display(filename1);
                                }
```

```
                        else {
                                fprintf(stderr, "Error in get_filename()₩n");
                        }
                        break;
                case COPY :
                        if (get_filename(filename1)) {
                                if (get_filename(filename2)) {
                                        f_copy(filename1, filename2);
                                }
                                else {
                                        fprintf(stderr, "Error in get_filename()₩n");
                                }
                        }
                        else {
                                fprintf(stderr, "Error in get_filename()₩n");
                        }
                        break;
                case APPEND :
                        if (get_filename(filename1)) {
                                if (get_filename(filename2)) {
                                        f_append(filename1, filename2);
                                }
                                else {
                                        fprintf(stderr, "Error in get_filename()₩n");
                                }
                        }
                        else {
                                fprintf(stderr, "Error in get_filename()₩n");
                        }
                        break;
                case QUIT :
                        end_of_program = 1;
                        break;
        }
    }
}
```

```
/*****************************************************************************/
void menu_out()
{
        printf("₩n₩n");
        printf("****************************************₩n");
        printf("*                                      *₩n");
        printf("* 1. display                           *₩n");
        printf("* 2. copy                              *₩n");
        printf("* 3. append                            *₩n");
        printf("* 4. quit                              *₩n");
        printf("*                                      *₩n");
        printf("****************************************₩n");
}

/*****************************************************************************/
int get_filename(char *pathname)
{
        printf("Enter filename : ");
        if (scanf("%s", pathname) == 1)
                return 1;
        else
                return 0;
}

/*****************************************************************************/
int f_display(const char *pathname)
{
        int  fd;
        char buf[BUFSIZE];
        int nread;

        if ((fd = open(pathname, O_RDONLY)) > 0) {
                while ((nread = read(fd, buf, BUFSIZE)) > 0) {  /* 괄호 주의 */
                        write(1, buf, nread);
                                /* BUFSIZE가 아닌 nread 임에 주의할 것! */
                                /* write의 결과가 nread인지 확인도 할 수 있음 */
                }
                close(fd);
                return (0);
        }
        else {
                fprintf(stderr, "cannot open %s", pathname);
                return (-1);
        }
}
```

```
/*****************************************************************************/
int f_copy(const char *original, const char *target)
{
        int fd1, fd2;
        char buf[BUFSIZE];
        int nread;

        if ((fd1 = open(original, O_RDONLY)) > 0) {
                if ((fd2 = open(target, O_WRONLY | O_CREAT, PERMS)) > 0) {
                        while ((nread = read(fd1, buf, BUFSIZE)) > 0) {
                                write(fd2, buf, nread);
                        }
                        close(fd2);
                        close(fd1);
                        return(0);
                }
                else {
                        fprintf(stderr, "cannot open or create %s", target);
                        close(fd1);
                        return(-1);
                }
        }
        else {
                fprintf(stderr, "cannot open %s", original);
                return(-1);
        }
}
```

```
/*****************************************************************************/
int f_append(const char *original, const char *target)
{
        int fd1, fd2;
        char buf[BUFSIZE];
        int nread;

        if ((fd1 = open(original, O_RDONLY)) > 0) {
                if ((fd2 = open(target, O_WRONLY | O_CREAT | O_APPEND, PERMS)) > 0) {
                        while ((nread = read(fd1, buf, BUFSIZE)) > 0) {
                                write(fd2, buf, nread);
                        }
                        close(fd2);
                        close(fd1);
                        return(0);
                }
                else {
                        fprintf(stderr, "cannot open or create %s", target);
                        close(fd1);
                        return (-1);
                }
        }
        else {
                fprintf(stderr, "cannot open %s", original);
                return(-1);
        }
}
```