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

#include <fcntl.h>
#include <unistd.h>

#define SEARCH  1
#define INSERT  2
#define DELETE  3
#define LIST    4
#define QUIT    5

#define LINE_LENGTH     255

#define NAME_LENGTH     24
#define PHONE_LENGTH    12

#define PERMS 0644

typedef struct {
        int     serial;
        char    name[NAME_LENGTH];
        char    phone[PHONE_LENGTH];
} name_card_t;

void menu_out();
int is_existing_serial(int fd, int no);
int get_by_serial(int fd);
int insert_by_serial(int fd);
int delete_by_serial(int fd);
void list_all(int fd);


/*****************************************************************************/
void main()
{
        int     end_of_program = 0;
        int     choice;
        int     fd;
        char *filename = "card_file";
        char line[LINE_LENGTH];

        if ((fd = open(filename, O_RDWR | O_CREAT, PERMS)) < 0) {
                fprintf(stderr, "cannot open %sWn", filename);
                exit(1);
```

```
        }

        while(!end_of_program){
                menu_out();

                printf("Enter your choice : ");
                if (scanf("%d", &choice) != 1) {
                        scanf("%s", line);
                        continue;
                }

                switch (choice) {
                        case SEARCH :
                                get_by_serial(fd);
                                break;
                        case INSERT :
                                insert_by_serial(fd);
                                break;
                        case DELETE :
                                delete_by_serial(fd);
                                break;
                        case LIST :
                                list_all(fd);
                                break;
                        case QUIT :
                                end_of_program = 1;
                }
        }
        close(fd);
}
```

```
/*****************************************************************************/
void menu_out()
{
        printf("₩n₩n");
        printf("****************************************₩n");
        printf("*                                       *₩n");
        printf("* 1. search by serial number            *₩n");
        printf("* 2. insert                             *₩n");
        printf("* 3. delete                             *₩n");
        printf("* 4. list all                           *₩n");
        printf("* 5. quit                               *₩n");
        printf("*                                       *₩n");
        printf("****************************************₩n");
}
```

```
/*****************************************************************************/
int is_existing_serial(int fd, int number)
{
        off_t   offset;

        offset = (number - 1) * sizeof(name_card_t);

        if (lseek(fd, offset, SEEK_SET) >= 0) {
                name_card_t     tmp_card;
                int     card_size;

                card_size = sizeof(tmp_card);
                memset(&tmp_card, 0, card_size);

                if (read(fd, &tmp_card, card_size) == card_size) {
                        if (tmp_card.serial == number) {
                                return (1);
                        }
                }
        }
        return (0);
}
```

```
/*****************************************************************************/
int get_by_serial(int fd)
{
        int number;
        off_t   last_pos;
        off_t   offset;
        name_card_t tmp_card;
        int card_size = sizeof(name_card_t);


        last_pos = lseek(fd, (off_t)0, SEEK_END);

        printf("Enter serial number : ");
        scanf("%d", &number);

        if ((number < 1) || (number > last_pos / card_size)) {
                fprintf(stderr, "범위를 벗어난 번호입니다.Wn");
                return(-1);
        }

        offset = (number-1) * sizeof(name_card_t);
        if (lseek(fd, offset, SEEK_SET) >= 0) {

                memset(&tmp_card, 0, card_size);
                if (read(fd, &tmp_card, card_size) == card_size) {
                        if (tmp_card.serial > 0) {
                                printf("Serial : %10dWt%30sWt%30sWn", tmp_card.serial,
                                                tmp_card.name, tmp_card.phone);
                                return (0);
                        }
                        else {
                                fprintf(stderr, "존재하지 않은 번호입니다.Wn");
                                return(-1);
                        }
                }
                else {
                        perror("read");
                        return(-1);
                }
        }
        else {
                perror("lseek");
                return(-1);
        }
}
```

```
/*****************************************************************************/
int insert_by_serial(int fd)
{
        int number = 0;
        off_t   offset;
        off_t   last_pos;
        name_card_t tmp_card;
        int card_size = sizeof(name_card_t);

        printf("Enter serial number : ");
        scanf("%d", &number);

        if (number < 1) {
                fprintf(stderr, "범위를 벗어난 번호입니다.₩n");
                return(-1);
        }

        last_pos = lseek(fd, (off_t)0, SEEK_END);
        offset = (number-1) * sizeof(name_card_t);

        if (lseek(fd, (off_t)offset, SEEK_SET) >= 0) {
                if (last_pos > offset) {
                        if (read(fd, &tmp_card, card_size) == card_size) {
                                if(tmp_card.serial == number) {
                                        fprintf(stderr, "사용 중인 번호입니다.₩n");
                                        return(-1);
                                }
                                if (lseek(fd, (off_t) -card_size, SEEK_CUR) < 0) {
                                        perror("lseek");
                                        return(-1);
                                }
                        }
                        else {
                                perror("read");
                                return(-1);
                        }
                }

                memset(&tmp_card, 0, card_size);
                tmp_card.serial = number;
                printf("Enter user name : ");
                scanf("%s", tmp_card.name);
                printf("Enter phone number : ");
                scanf("%s", tmp_card.phone);

                if (write(fd, &tmp_card, card_size) == card_size) {
```

```
                        return(0);
                }
                else {
                        perror("lseek");
                        return(-1);
                }
        }
        perror("lseek");
        return(-1);
}
```

```
/*******************************************************************************/
int delete_by_serial(int fd)
{
        int number;
        off_t   last_pos;
        off_t   offset;
        name_card_t tmp_card;
        int card_size = sizeof(name_card_t);


        last_pos = lseek(fd, (off_t)0, SEEK_END);

        printf("Enter serial number : ");
        scanf("%d", &number);

        if ((number < 1) || (number > last_pos / card_size)) {
                fprintf(stderr, "범위를 벗어난 번호입니다.₩n");
                return(-1);
        }

        offset = (number-1) * sizeof(name_card_t);
        if (lseek(fd, offset, SEEK_SET) >= 0) {
                if (read(fd, &tmp_card, card_size) == card_size) {
                        if(tmp_card.serial == number) {
                                printf("%d 카드를 삭제합니다.₩n", number);
                                memset(&tmp_card, 0, card_size);
                                lseek(fd, (off_t) -card_size, SEEK_CUR);
                                write(fd, &tmp_card, card_size);
                                return(0);
                        }
                        else {
                                fprintf(stderr, "등록하지 않은 번호입니다.₩n");
                                return(-1);
                        }
                }
                perror("read");
                return(-1);
        }
        else {
                perror("lseek");
                return(-1);
        }
}
```

```
/*******************************************************************************/
void list_all(int fd)
{
        name_card_t tmp_card;
        int card_size = sizeof(name_card_t);

        lseek(fd, (off_t)0, SEEK_SET);

        memset(&tmp_card, 0, card_size);
        while (read(fd, &tmp_card, card_size) == card_size) {
                if (tmp_card.serial > 0) {
                        printf("%10d:Wt%30sWt%30sWn", tmp_card.serial,
                                        tmp_card.name, tmp_card.phone);
                }
                memset(&tmp_card, 0, card_size);
        }
}
```