

## 1. 소스코드

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

typedef struct Score{
    int count[3];
    char name[30];
    char grade;
    char address[50];
    int price;
    char delete;
} Score;

void readScore(Score s) {
    printf("%4d-%d-%d %6s %4c %7d %5s\n", s.count[0], s.count[1], s.count[2], s.name, s.grade, s.price, s.address);
}

void listMenu() {
    printf("\n*** 메뉴판 ***\n");
    printf("1. 비빔밥   8,000 원\n2. 김치찌개 7,000 원\n3. 참치덮밥 9,000 원\n*****\n");
}

void listOrder(Score *s, int count) {
    printf("\nNo   메뉴개수   예약자   등급   가격   주소\n=====");
    int a=count;
    for (int i=0; i<count; i++) {
        if (s[i].count[0]==-1) {
            a--;
            continue;
        }
        printf("%2d", i+1);
        readScore(s[i]);
    }
    printf("\n 총 %d 개의 주문이 있습니다.\n", a);
}

int selectDataNo (Score *s, int count) {
    int no;
    listOrder(s, count);
    printf("번호는 (취소 : 0)? ");
    scanf("%d", &no);
    return no;
}
```

```

int addOrder(Score *s) {
    printf("\n 메뉴별 개수 ? ");
    scanf("%d %d %d", &s->count[0], &s->count[1], &s->count[2]);
    printf("예약자는 ? ");
    scanf(" %s", s->name);
    printf("예약자 등급은(B,S,G) ? ");
    scanf(" %c", &s->grade);
    printf("배달 주소는 ? ");
    scanf(" %[^\n]s", s->address);
    s->price=8000*s->count[0]+7000*s->count[1]+9000*s->count[2];
    printf("총 주문한 음식 값은 %d 입니다.\n", s->price);
    return 1;
}

// int updateScore(Score* s) {
//     printf("새 메뉴명은 ? ");
//     scanf(" %[^\n]s", s->menu);
//     printf("새 메뉴종류(P/S/R)? ");
//     scanf(" %c", &s->type);
//     printf("새 가격은 ? ");
//     scanf(" %d", &s->price);
//     printf("=>수정성공!\n");
//     return 1;
// }

int deleteMenu(Score *s) {
    s->count[0]=-1;
    printf("=> 삭제됨!\n");
    return 1;
}

void saveData(Score *sp, int count) {
    FILE *fp;
    fp=fopen("orders.txt", "wt");
    for (int i=0; i< count; i++) {
        if(sp[i].count[0]==-1) continue;
        fprintf(fp, "%d-%d-%d %s %c %d %s\n", sp[i].count[0], sp[i].count[1], sp[i].count[2], sp[i].name, sp[i].grade,
sp[i].price, sp[i].address);
    }
    fclose(fp);
    printf("=> 저장됨! \n");
}

int loadData(Score *sp) {
    int count=0, i=0;
    FILE *fp;

```

```

fp=fopen("orders.txt", "rt");
printf("=> 데이터 가져오는 중...\n");
if (fp==NULL) {
    printf("=> 파일 없음\n");
}
else {
    for (;i<30;i++) {
        fscanf(fp, "%d", &sp[i].count[0]);
        if (feof(fp)) break;
        fscanf(fp, "%d", &sp[i].count[1]);
        fscanf(fp, "%d", &sp[i].count[2]);
        fscanf(fp, " %s", sp[i].name);
        fscanf(fp, " %c", &sp[i].grade);
        fscanf(fp, " %d", &sp[i].price);
        fscanf(fp, "%[^\n]s", sp[i].address);
        count++;
    }
    printf("=> %d 개의 주문 정보를 읽었습니다!\n", count);
}
fclose(fp);
return i;
}

void findname(Score *s, int count) {
    int scnt=0;
    char search[20];
    printf("검색할 이름? ");

    scanf("%s", search);
    printf("\nNo   메뉴개수   예약자   등급   가격   주소\n=====");
    for (int i=0; i<count; i++) {
        if (s[i].count[0]==-1) continue;
        if (strstr(s[i].name, search)) {
            printf("%2d", i+1);
            readScore(s[i]);
            scnt++;
        }
    }
    printf("\n 총 %d 개의 주문이 있습니다.\n", scnt);
    if (scnt==0) printf("=> 검색된 데이터 없음!\n");
}

int selectMenu(){
    int menu;
    printf("\n*** 맛있는 한식 배달 ***\n1. 조회\n2. 추가\n3. 수정(기능구현하지 않음)\n4. 삭제\n5. 저장\n6.
예약자검색\n0. 종료\n");
    printf("\n=> 원하는 메뉴는? ");

```

```

scanf("%d", &menu);
return menu;
}

int main(void){
    Score sp[30];
    int count = 0, menu;
    int index=0;

    count = loadData(sp);
    index = count;
    while (1){
        menu = selectMenu();
        if (menu == 0) break;
        if (menu == 1 || menu == 3 || menu == 4) {
            if (count==0) continue;
        }
        if (menu == 1) {
            listOrder(sp, index);
        }
        else if (menu == 2) {
            listMenu();
            count += addOrder(&sp[index++]);
        }
        // else if (menu == 3) {
        //     int no=selectDataNo(sp, index);
        //     if (no==0) {
        //         printf("=> 취소됨!\n");
        //         continue;
        //     }
        //     updateScore(&sp[no-1]);
        // }
        else if (menu == 4) {
            int no=selectDataNo(sp, index);
            if (no==0) {
                printf("=> 취소됨!\n");
                continue;
            }
            int deleteok;
            printf("정말로 삭제하시겠습니까? (삭제 : 1)");
            scanf("%d", &deleteok);
            if (deleteok == 1) {
                if (deleteMenu(&sp[no-1])) {
                    count--;
                }
            }
        }
    }
}

```

```

    else if (menu==5) {
        saveData(sp, index);
    }
    else if (menu==6) {
        findname(sp, index);
    }
}
printf("종료 됨!\n");
return 0;
}

```

## 2. 결과

\*\*\* 맛있는 한식 배달 \*\*\*

1. 조회
2. 추가
3. 수정
4. 삭제
5. 저장
6. 예약자검색
0. 종료

=> 원하는 메뉴는? 2

\*\*\*\* 메뉴판 \*\*\*\*

1. 비빔밥 8,000 원
2. 김치찌개 7,000 원
3. 참치덮밥 9,000 원

\*\*\*\*\*

메뉴별 개수 ? 1 1 1

예약자는? jang

예약자 등급은(B,S,G) ? B

배달 주소는 ? 양덕 트리니언

총 주문한 음식 값은 24000 입니다.

\*\*\* 맛있는 한식 배달 \*\*\*

1. 조회
2. 추가
3. 수정
4. 삭제
5. 저장
6. 예약자검색
0. 종료

=> 원하는 메뉴는? 2

\*\*\*\* 메뉴판 \*\*\*\*

1. 비빔밥 8,000 원
  2. 김치찌개 7,000 원
  3. 참치덮밥 9,000 원
- \*\*\*\*\*

메뉴별 개수 ? 2 1 2

예약자는? sally

예약자 등급은(B,S,G) ? S

배달 주소는 ? 장성 법원

총 주문한 음식 값은 41000 입니다.

=> 원하는 메뉴는? 1

No 메뉴개수 예약자 등급 가격 주소

=====

1 1-1-1 jang B 24000 양덕 트리니언

2 2-1-2 sally S 41000 장성 법원

총 2 개의 주문이 있습니다.

\*\*\* 맛있는 한식 배달 \*\*\*

1. 조회
2. 추가
3. 수정

- 4. 삭제
- 5. 저장
- 6. 예약자검색
- 0. 종료

=> 원하는 메뉴는? 4

No 메뉴개수 예약자 등급 가격 주소

=====

- 1 1-1-1 jang 0 B 24000 양덕 트리니엔
- 2 2-1-2 sally 0 S 41000 장성 법원

총 2 개의 주문이 있습니다.

번호는? (취소:0)? 2

정말로 삭제하시겠습니까?(삭제:1) 1

=> 삭제됨!

\*\*\* 맛있는 한식 배달 \*\*\*

- 1. 조회
- 2. 추가
- 3. 수정
- 4. 삭제
- 5. 저장
- 6. 예약자검색
- 0. 종료

=> 원하는 메뉴는? 1

No 메뉴개수 예약자 등급 가격 주소

=====

- 1 1-1-1 jang 0 B 24000 양덕 트리니엔

총 1 개의 주문이 있습니다.

\*\*\* 맛있는 한식 배달 \*\*\*

1. 조회
2. 추가
3. 수정
4. 삭제
5. 저장
6. 예약자검색
0. 종료

=> 원하는 메뉴는? 6

검색할 이름? Lee

No 메뉴개수 예약자 등급 가격 주소

=====

총 0 개의 주문이 있습니다.

=> 검색된 데이터 없음!

\*\*\* 맛있는 한식 배달 \*\*\*

1. 조회
2. 추가
3. 수정
4. 삭제
5. 저장
6. 예약자검색
0. 종료

=> 원하는 메뉴는? 6

검색할 이름? jang

No 메뉴개수 예약자 등급 가격 주소

=====

1 1-1-1 jang 0 B 24000 양덕 트리니언



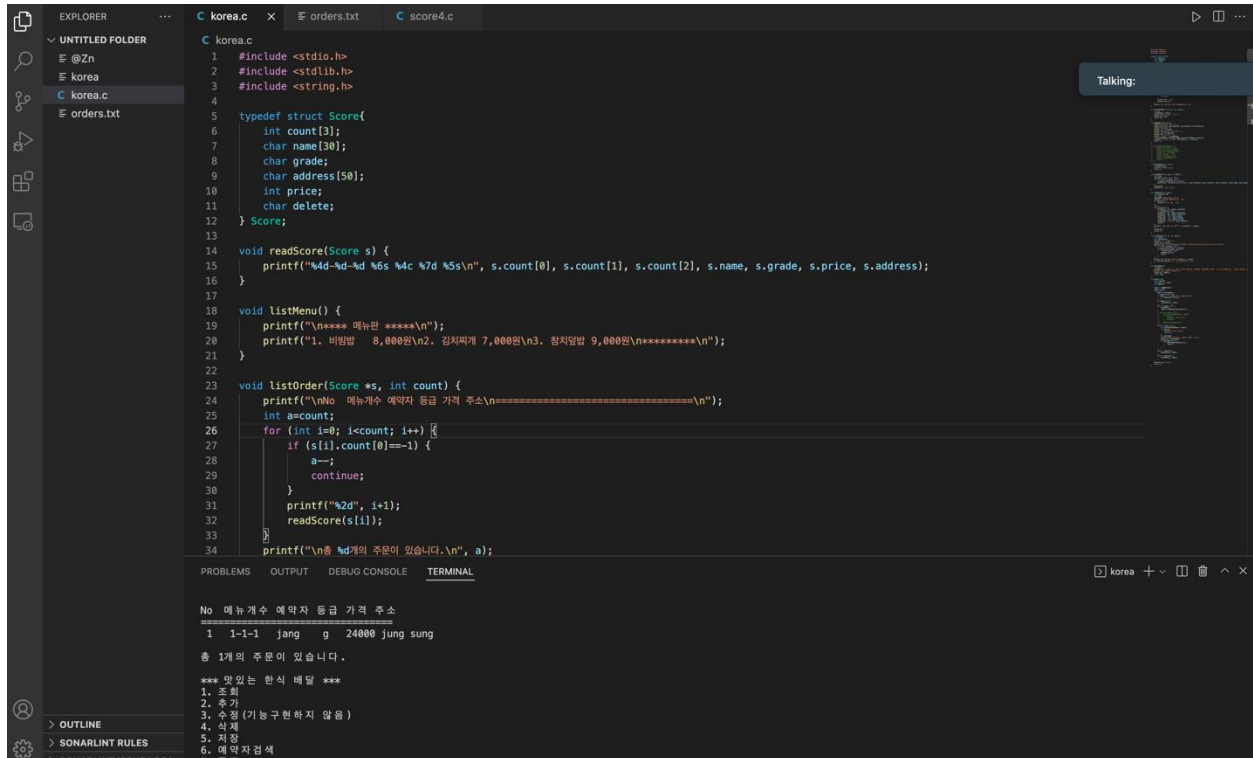
총 1 개의 주문이 있습니다.

### 3. orders.txt

1-1-1 jang B 24000 양덕 트리니엔

2-1-2 sally S 41000 장성 법원

### 4.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 typedef struct Score{
6     int count[3];
7     char name[30];
8     char grade;
9     char address[50];
10    int price;
11    char delete;
12 } Score;
13
14 void readScore(Score s) {
15     printf("%4d-%d-%d %6s %4c %7d %5s\n", s.count[0], s.count[1], s.count[2], s.name, s.grade, s.price, s.address);
16 }
17
18 void listMenu() {
19     printf("\n*** 메뉴판 ***\n");
20     printf("1. 비빔밥    8,000원\n2. 김치찌개 7,000원\n3. 참치덮밥 9,000원\n*****\n");
21 }
22
23 void listOrder(Score *s, int count) {
24     printf("\nNo    메뉴개수    예약자    등급    가격    주소\n");
25     int a=count;
26     for (int i=0; i<count; i++) {
27         if (s[i].count[0]==-1) {
28             a--;
29             continue;
30         }
31         printf("%2d", i+1);
32         readScore(s[i]);
33     }
34     printf("\n총 %d개의 주문이 있습니다.\n", a);
35 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
No    메뉴개수    예약자    등급    가격    주소
1    1-1-1    jang    g    24000    jung sung

총 1개의 주문이 있습니다.

*** 맛있는 한식 배달 ***
1. 조회
2. 추가
3. 수정 (기능구현하지 않음)
4. 삭제
5. 저장
6. 예약자 검색
0. 종료
```

### 5. 완성한 기능

1. 조회 기능

2. 추가 기능

3. 삭제 기능

5. 저장 기능 및 불러오기 기능 (저장 기능까지는 했지만 불러오기 기능에서 오류가 살짝 있는 것 같습니다. 선택 추가 문제이기에 그냥 제출 하겠습니다!)

6. 예약자 검색 기능

0. 종료 기능

