

C프로그래밍및실습

# 복합쇼핑몰

최종 보고서

제출일자: 2023-12-24

제출자명: 강지효

제출자학번: 232676

## 1. 프로젝트 목표

### 1) 배경 및 필요성

복합쇼핑몰이란 영화상영관, 쇼핑센터, 식당 등을 한 건물 내에 갖춘 복합건물을 말한다. 산업이 발전하면서 사람들의 취미가 늘어났다. 다양한 취미를 하나의 건물에서 해결해주기 위한 복합쇼핑몰이 차차 늘어나며 현재는 많은 복합쇼핑몰이 있다. 더욱 복잡해진 복합쇼핑몰에는 안내자가 필요하기 마련이다. 그래서 복합쇼핑몰에 들어오자마자 모든 것을 키오스크로 간단히 해결해주는 프로그램이 필요하다.

### 2) 프로젝트 목표

복합쇼핑몰의 기능이 무엇이 있는지를 분석하고 고객이 원하는 취미활동을 입력받아 위치, 가격, 예매(영화)등의 다양한 기능을 입장할 때의 키오스크를 통해 미리 해결하는 것이 목표이다.

### 3) 차별점

기존에는 각각의 가게에 직접 가서 모든 것을 해결해야 했다면, 건물에 들어가자마자 키오스크를 통해 문제를 해결할 수 있으니 시간을 단축할 뿐만 아니라 위치를 알려주기 때문에 복잡한 건물 안에서 길을 잃거나 그로 인해 약속시간에 늦을 일이 없도록 도와준다.

## 2. 기능 계획

### 1) 고객이 이곳에 온 목적을 파악하고 이를 도와주는 기능

- 고객이 이곳에 온 목적을 알아내어 기능을 실행한다.

#### (1) 영화관 예매 기능

- 영화 장르를 고를 수 있고 그에 맞는 영화와 상영 시간을 제시하여 사용자에게 선택하게 하여 그 영화를 예매해주는 프로그램

## (2) 쇼핑 할 곳의 위치 정보 제공 기능

- 식료품, 의류, 명품의 카테고리로 나눠 각 카테고리의 브랜드를 알려주고 그 브랜드를 선택할 경우 위치를 알려주는 프로그램.

## (3) 식당에서의 메뉴 주문 프로그램

- 한식, 일식, 중식으로 카테고리를 나누고 이에 맞는 메뉴와 가격을 알려주고 고객에게 선택하도록 하는 프로그램.

## (4) 운동 할 곳의 위치를 알려주는 프로그램

- 헬스, 축구, 테니스로 카테고리를 나누고 각각의 위치를 알려줌.

## 2) 주차 예약 기능

- 주차장 이용 시 주차 비용과 주차 가능한 층 수 정보 제공.

### 1) 층 수와 주차하지 못하는 장소를 보여주고 주차 예약 할 수 있는 기능.

고객이 차를 가지고 왔는지를 물어보고 주차 공간 정보 제공 및 자리 예약.

-> 이렇게 하면 주차장 내에서 주차 자리가 없어 주차 자리를 찾는 시간을 줄일 수 있을 것이라고 생각.

## 3) 일행 파악 기능

- 일행이 있다면 일행에게 자신이 들어왔음을 알려주는 기능.

### 1) 일행 번호 입력 시 자신의 일행이 들어와서 번호 입력 시 문자 전송

들어가 있는 사람에게 만나기 쉽게 핸드폰 문자로 일행이 들어온 것을 알림. 사람이 들어갈 때 일행이 있는지를 물어보고, 있다면 일행의 핸드폰 번호를 입력하

게 한다. 그리고 나중에 그 일행이 들어올 때 자신의 핸드폰 번호를 입력하게 될 경우 먼저 들어간 사람에게는 알림이 오게 된다

### 3. 기능 구현

#### (1) 고객이 이곳에 온 목적을 파악하고 이를 도와주는 기능

##### 영화관 예매 기능

###### - 입력

1. 보고 싶은 영화 장르 번호 입력.
2. 보고 싶은 영화 번호 입력.
3. 원하는 영화 시간 번호 입력
4. 시간이 맞는지 확인 하는 절차에 "예", "아니오" 입력

###### - 출력

1. 영화 장르 출력
2. 영화 장르에 맞는 제목 출력
3. 해당 영화의 시간표 출력
4. 시간이 맞는지 확인
5. 예를 입력 시 시간과 영화를 알려주며 함수 끝
6. 아니오를 입력 시 시간을 다시 골라 달라며 다시 시작
7. 이외의 답변을 입력 시 다시 예, 아니오를 입력하도록 유도

- 설명 : 영화관 키오스크와 같이 영화, 시간표를 제시하고 자신이 원하는 시간대를 선택하여 예매 할 수 있게 함.

고객의 목적을 도와주는 기능을 하나의 헤더함수로 정리

###### - 적용된 배운 내용

헤더 함수를 만들기 위해 num1\_function.h 파일을 만들고 그에 대한 기능을 구현하기 위

해 num1\_function.c 파일에 아래에 설명할 함수들을 모두 구현함.

```
void movie_theater();
void select_movie(char** time_movie);
void select_movie_time(char** time_movie, int want_time, char* select_time);
void shopping();
typedef struct Korean_Food_price {
    char food_name[3][10];
    int food_price[3];
}Kfood_price;

typedef struct Japanese_Food_price {
    char food_name[3][10];
    int food_price[3];
}Jfood_price;

typedef struct Chinese_Food_price {
    char food_name[3][10];
    int food_price[3];
}Cfood_price;
void select_korean_menu(Kfood_price food, int yes_or_no, int food_number);
void select_japanese_menu(Jfood_price food, int yes_or_no, int food_number);
void select_chinese_menu(Cfood_price food, int yes_or_no, int food_number);
void cafeteria();
typedef struct exercise_kind {
    char type[3][10];
    char exercise_place[3][10];
    char* place[3];
}Exerkind;
void initialize(Exerkind* exercise_kind);
void exercise();
```

- 아래의 movie\_theater, shopping, cafeteria, exercise 함수들을 모두 num1\_function.c 파일에 구현함.

영화관 기능을 수행하는 movie\_theater 함수

- 적용된 배운 내용

반복문을 이용하여 기능이 계속 이어지도록 만들었음. 번호 4를 입력했을 시 반복문을 나가도록 설정

조건문을 이용하여 movie\_number에 저장된 번호에 따라 다른 내용이 출력되도록 설정  
2차원 배열을 이용하여 영화 상영 시간을 문자열 형태로 저장.

```
void movie_theater() { // 영화를 선택했을 시의 메인에 오는 함수.
    int genre;
    char time_elemental[9][9] = { "10시20분", "11시00분", "11시40분", "12시40분", "13시30분", "14시20분", "15시30분", "16시10분", "17시00분" };
    char time_frozen_kingdom[9][9] = { "10시30분", "11시10분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };
    char time_minions[9][9] = { "10시40분", "11시20분", "12시00분", "12시40분", "13시30분", "14시20분", "15시20분", "16시10분", "17시00분" };
    char time_avengers_endgame[9][9] = { "10시20분", "11시00분", "11시40분", "12시40분", "13시30분", "14시20분", "15시20분", "16시10분", "17시00분" };
    char time_guardians_of_galaxy[9][9] = { "10시30분", "11시10분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };
    char time_batman[9][9] = { "10시40분", "11시20분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };
    char time_captain_america_winter_soldier[9][9] = { "10시20분", "11시10분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };
    char time_the_roundup[9][9] = { "10시30분", "11시10분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };
    char time_kingsman[9][9] = { "10시40분", "11시20분", "11시50분", "12시50분", "13시40분", "14시30분", "15시30분", "16시20분", "17시10분" };

    while (1) {
        printf("보고싶은 장르를 골라주세요\n 1. 액션\n 2. 히어로\n 3. 액션\n 4. 종류 : ");
        scanf("%d", &genre);
        if (genre == 4) {
            break;
        }
        while (1) {
            if (genre == 1) {
                int movie_number;
                printf("무엇을 보고 싶으신가요?\n 1. 엘리멘탈\n 2. 겨울왕국\n 3. 미니언즈 : ");
                scanf("%d", &movie_number);
                printf("\n");
                if (movie_number == 1) {
                    select_movie(time_elemental);
                    break;
                }
                else if (movie_number == 2) {
                    select_movie(time_frozen_kingdom);
                    break;
                }
                else if (movie_number == 3) {
                    select_movie(time_minions);
                    break;
                }
                else {
                    printf("다시 선택해주세요.");
                }
            }
            else if (genre == 2) {
                int movie_number;
                printf("무엇을 보고 싶으신가요?\n 1. 어벤져스 엔드게임\n 2. 가디언즈 오브 갤럭시 3\n 3. 배트맨 : ");
                scanf("%d", &movie_number);
                printf("\n");
                if (movie_number == 1) {
                    select_movie(time_avengers_endgame);
                    break;
                }
                else if (movie_number == 2) {
                    select_movie(time_guardians_of_galaxy);
                    break;
                }
                else if (movie_number == 3) {
                    select_movie(time_batman);
                    break;
                }
                else {
                    printf("다시 선택해주세요.");
                }
            }
            else if (genre == 3) {
                int movie_number;
                printf("무엇을 보고 싶으신가요?\n 1. 캡틴 아메리카 윈터솔저\n 2. 범죄도시2\n 3. 킹스맨 : ");
                scanf("%d", &movie_number);
                printf("\n");
                if (movie_number == 1) {
                    select_movie(time_captain_america_winter_soldier);
                    break;
                }
                else if (movie_number == 2) {
                    select_movie(time_the_roundup);
                    break;
                }
                else if (movie_number == 3) {
                    select_movie(time_kingsman);
                    break;
                }
                else {
                    printf("다시 선택해주세요.");
                }
            }
        }
    }
}
//1번 함수 만들기 끝
```

movie\_theater 함수에서 영화를 선택하는 기능인 select\_movie 함수

- 적용된 배운 내용

배열 포인터에 함수에서 매개변수로 받은 2차원 배열 time\_movie를 저장.

이를 이용하여 시간표를 출력함

strcpy\_s 함수를 이용하여 select\_time 변수에 원하는 시간 문자열 형태로 저장.

```
void select_movie(char** time_movie) { //함수화 시켜서 하나로 다 사용하기. movie_theater의 세부 함수
    char(*p)[9] = time_movie; //배열 포인터에 저장 완료
    int want_time;
    int random;
    char select_time[10];
    char answer[10];
    char c;
    for (int i = 0; i < 9; i++) {
        //인덱스에 맞는 숫자부터 출력됨. 하지만 그 이후에 계속 나오고 한자가 나옴.
        //문자열에 띄어쓰기가 포함되어 있어서 그랬음. 왜 그러는지 질문
        printf("%d. %s\n", i + 1, p[i]);
    }
    printf("위의 시간표에서 시간을 골라주세요(숫자 작성) : ");
    scanf_s("%d", &want_time);
    strcpy_s(select_time, sizeof(p[want_time - 1]), p[want_time - 1]); //select_time에 p[want_time]시간 대입.
    select_movie_time(time_movie, want_time, select_time);
}
```

select\_movie 함수에서 영화 시간을 고르는 기능을 하는 select\_movie\_time 함수

- 적용된 배운 내용

함수에서 매개변수로 2차원 배열을 받아 배열 포인터에 저장.

while~ 반복문을 사용하여 계행 문자를 삭제

do-while 반복문을 이용하여 시간이 맞는지를 일단 한번 무조건 실행하도록 유도 조건은

answer 문자열 변수가 "예"가 된다면 종료하도록 함.

strcmp 함수를 이용하여 answer와 "예", "아니오"가 같은 문자열 인지 검사하도록 하여

조건문에 적용.

```
//1번 함수 만들기 시작
void select_movie_time(char** time_movie, int want_time, char* select_time) { //select_movie의 세부 함수
    char(*p)[9] = time_movie;
    char answer[10];
    char c;
    do {
        if (want_time <= 9 && want_time >= 1) {
            printf("%s이 선택한 시간이 맞습니까? (예, 아니오로 대답해주세요) : ", select_time);
            scanf_s("%s", answer, sizeof(answer)); //answer에 대답 저장 완료 하지만 엔터까지 answer 변수에 저장되는 듯. 이걸 해결해야 할 듯.
            while ((c = getchar()) != '\n' && c != EOF); //계행 문자 삭제
            if (strcmp(answer, "예") == 0) { //계행 문자가 포함되면서 오류 발생
                printf("%s으로 예매 완료 되었습니다. 시작 10분전에 입장해주세요.\n", select_time);
                break;
            }
            else if (strcmp(answer, "아니오") == 0) {
                printf("그렇다면 원하는 시간을 다시 골라주세요.\n");
                printf("위의 시간표에서 시간을 골라주세요(숫자 작성) : ");
                scanf_s("%d", &want_time);
                strcpy_s(select_time, sizeof(p[want_time - 1]), p[want_time - 1]); //select_time에 p[want_time]시간 대입.
                continue;
            }
            else {
                printf("답변을 다시 선택해주세요.\n");
            }
        }
    } while (answer != "예");
}
```

## 쇼핑 할 곳의 위치 정보 제공 기능

### - 입력

1. 쇼핑할 품목에 대하여 번호 입력
2. 의류, 명품 입력 시 원하는 브랜드 입력
3. 브랜드 번호 입력 시 브랜드의 위치 정보 제공

### - 출력

1. 쇼핑할 품목 출력
2. 브랜드 종류 출력
3. 브랜드 위치 출력
4. 브랜드 이름, 위치를 한 마디로 안내

- 설명 : 쇼핑의 목적을 물어보고 그에 맞는 매장의 위치를 알려주는 기능이다.

## 쇼핑 할 곳의 위치 정보 제공 기능을 수행하는 shopping 함수

### - 적용된 배운 내용

2차원 배열을 초기화 한 후 배열 포인터를 사용하여 2차원 배열 저장.

반복문 while을 무한루프를 통해 계속 키오스크를 사용 할 수 있게 제작 4번을 입력 시 반복문을 빠져나가게 설정.

조건문을 사용하여 purpose\_shopping 변수의 값에 따라 실행 되는 문장을 다르게 설정.

문자열 함수 strcpy\_s를 이용하여 brand\_name 변수에 브랜드 이름 저장.



```

void shopping() {
    int purpose_shopping;
    char market_place[9][9] = { "나이키", "스파이더", "8seconds", "MLB", "FILA", "유니클로", "스투시", "무신사", "아디다스" };
    char clothes_brand[9][9] = { "스파이더", "나이키", "아디다스", "MLB", "8seconds", "유니클로", "FILA", "무신사", "스투시" };
    char luxury_brand[9][22] = { "CHANEL\0", "DIOR\0", "BALENCIAGA\0", "GUCCI\0", "Yves Saint Laurent\0", "PRADA\0", "Louis Vuitton\0", "Hermes\0", "Burberry\0" };
    char(*p)[9] = clothes_brand;
    char(*q)[22] = luxury_brand;
    char brand_name[20];
    int brand_number;
    int j;
    printf("쇼핑할 품목이 무엇인가요?\n 1. 식자재 2. 의류 3. 명품 4. 종류: ");
    scanf_s("%d", &purpose_shopping);
    while (1) {
        if (purpose_shopping == 1) {
            printf("지하 1층에 이마트가 있습니다.\n\n");
        }
        else if (purpose_shopping == 2) {
            printf("어떤 브랜드를 원하십니까?\n 1. 스파이더 2. 나이키 3. 아디다스 4. MLB 5. 8seconds 6. 유니클로 7. FILA 8. 무신사 9. 스투시 : ");
            scanf_s("%d", &brand_number);
            strcpy_s(brand_name, sizeof(brand_name), p[brand_number - 1]);
            for (int i = 0; i <= 8; i++) {
                printf("%d. %s", i + 1, p[i]);
                printf("\t");
                if (i == 3 || i == 6) {
                    printf("\t");
                }
                if (i == 2 || i == 5) {
                    printf("\n");
                }
            }
            printf("\n");
            printf("%s는 위의 표의 %d번에 있습니다.\n\n", brand_name, brand_number);
        }
        else if (purpose_shopping == 3) {
            printf("어떤 브랜드를 원하십니까?\n 1. CHANEL 2. DIOR 3. BALENCIAGA 4. GUCCI 5. Yves Saint Laurent 6. PRADA 7. Louis Vuitton 8. Hermes 9. Burberry : ");
            scanf_s("%d", &brand_number);
            strcpy_s(brand_name, sizeof(brand_name), q[brand_number - 1]);
            for (int i = 0; i <= 8; i++) {
                printf(" %s ", q[i]); //3,5,7번 입력시 오류 다른 파일에 실행해보니 완벽하게 작동.
                if (i == 0) {
                    printf("\t");
                }
                if (i == 1) {
                    printf("\t");
                }
                if (i == 2 || i == 5) {
                    printf("\n");
                }
            }
            printf("\n");
            printf("%s는 위의 표의 %d번에 있습니다.\n\n", brand_name, brand_number);
        }
        else if (purpose_shopping == 4) {
            break;
        }
        printf("쇼핑할 품목이 무엇인가요?\n 1. 식자재 2. 의류 3. 명품 4. 종류: ");
        scanf_s("%d", &purpose_shopping);
    }
}

```

## 식당에서의 메뉴 주문 프로그램

### - 입력

1. 먹고 싶은 음식의 종류 번호 입력
2. 먹고 싶은 음식 번호 입력
3. 메뉴 확인 번호 입력

### - 출력

1. 음식의 종류 출력

2. 각 카테고리에 맞는 메뉴 출력
3. 메뉴 맞게 골랐는지 확인
4. 맞을 시 가격과 메뉴 한 마디로 출력
5. 아닐 시 메뉴를 다시 고르도록 유도
6. 잘못된 답을 입력 시 다시 입력하도록 유도

- 설명 : 자신이 원하는 종류의 음식을 고른 후에 가격과 함께 주문이 들어가는 기능.

## 식당에서의 메뉴 주문 프로그램을 수행하는 cafeteria 함수

### - 적용된 배운 기능

구조체를 이용하여 각 국의 음식 이름, 가격을 저장.

반복문을 사용하여 kind\_of\_food 변수에 저장된 숫자에 따라 다른 기능을 실행 하도록

설정

각 국의 음식 메뉴 고르는 함수를 다 따로 만들어서 사용

```
void cafeteria() {
    //구조체와 배열 저장할때
    Kfood_price kfood = { {"제육볶음"}, {"김치찌개"}, {"동순비빔밥"} }, {7000, 6000, 6000} };
    Jfood_price jfood = { {"동심카즈", "마제소바", "모듬초밥"} }, {9000, 9000, 15000} };
    Cfood_price cfood = { {"짜장면", "관동", "탕수육(소)"} }, {5000, 7000, 10000} };
    int kind_of_food;
    printf("어떤 종류의 음식을 드시고 싶으신가요?\n 1. 한식 2. 일식 3. 중식 : ");
    scanf_s("%d", &kind_of_food);
    int food_number;
    int yes_or_no;
    if (kind_of_food == 1) {
        for (int i = 0; i < 3; i++) {
            printf("%d. %s - %d원 ", i + 1, kfood.food_name[i], kfood.food_price[i]);
        }
        printf("\n");
        printf("위의 메뉴에서 하나 골라주세요 : ");
        scanf_s("%d", &food_number);
        printf("고르신 메뉴가 %s(0)가 맞습니까?\t 1. 예 2. 아니요 : ", kfood.food_name[food_number - 1]);
        scanf_s("%d", &yes_or_no);
        select_korean_menu(kfood, yes_or_no, food_number);
    }
    if (kind_of_food == 2) {
        for (int i = 0; i < 3; i++) {
            printf("%d. %s - %d원 ", i + 1, jfood.food_name[i], jfood.food_price[i]);
        }
        printf("\n");
        printf("위의 메뉴에서 하나 골라주세요 : ");
        scanf_s("%d", &food_number);
        printf("고르신 메뉴가 %s(0)가 맞습니까?\t 1. 예 2. 아니요 : ", jfood.food_name[food_number - 1]);
        scanf_s("%d", &yes_or_no);
        select_japanese_menu(jfood, yes_or_no, food_number);
    }
    if (kind_of_food == 3) {
        for (int i = 0; i < 3; i++) {
            printf("%d. %s - %d원 ", i + 1, cfood.food_name[i], cfood.food_price[i]);
        }
        printf("\n");
        printf("위의 메뉴에서 하나 골라주세요 : ");
        scanf_s("%d", &food_number);
        printf("고르신 메뉴가 %s(0)가 맞습니까?\t 1. 예 2. 아니요 : ", cfood.food_name[food_number - 1]);
        scanf_s("%d", &yes_or_no);
        select_chinese_menu(cfood, yes_or_no, food_number);
    }
}
//4번 함수 만들기 끝.
```

```

//3번 함수 만들기 시작
typedef struct Korean_Food_price {
    char food_name[3][10];
    int food_price[3];
}Kfood_price;

typedef struct Japanese_Food_price {
    char food_name[3][10];
    int food_price[3];
}Jfood_price;

typedef struct Chinese_Food_price {
    char food_name[3][10];
    int food_price[3];
}Cfood_price;

```

각 국의 음식 메뉴를 고르는 기능을 수행하는 select\_(국가)\_menu

- 적용된 배운 내용

구조체를 매개변수로 받아 함수 내에서 사용

do-while문을 이용하여 반복문 진행 조건 없이 무한루프를 사용하여 반복 yes\_or\_no 변수 값이 1이 됐을 때 반복문 탈출.

```

void select_korean_menu(Kfood_price food, int yes_or_no, int food_number) {
    do {
        if (yes_or_no == 1) {
            printf("%s %d원입니다.\n", food.food_name[food_number - 1], food.food_price[food_number - 1]);
            break;
        }
        else if (yes_or_no == 2) {
            printf("그렇다면 메뉴를 다시 골라주세요.\n");
            for (int i = 0; i < 3; i++) {
                printf("%d. %s - %d원 ", i + 1, food.food_name[i], food.food_price[i]);
            }
            printf("\n");
            printf("위의 메뉴에서 하나 골라주세요 : ");
            scanf_s("%d", &food_number);
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
            continue;
        }
        else {
            printf("잘못된 입력입니다. 다시 입력해주세요.\n");
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
        }
    } while (1);
}

```

```

void select_japanese_menu(Jfood_price food, int yes_or_no, int food_number) {
    do {
        if (yes_or_no == 1) {
            printf("%s %d원입니다.\n", food.food_name[food_number - 1], food.food_price[food_number - 1]);
            break;
        }
        else if (yes_or_no == 2) {
            printf("그렇다면 메뉴를 다시 골라주세요.\n");
            for (int i = 0; i < 3; i++) {
                printf("%d. %s - %d원 ", i + 1, food.food_name[i], food.food_price[i]);
            }
            printf("\n");
            printf("위의 메뉴에서 하나 골라주세요 : ");
            scanf_s("%d", &food_number);
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
            continue;
        }
        else {
            printf("잘못된 입력입니다. 다시 입력해주세요.\n");
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
        }
    } while (1);
}

```

```

void select_chinese_menu(Cfood_price food, int yes_or_no, int food_number) {
    do {
        if (yes_or_no == 1) {
            printf("%s %d원입니다.\n", food.food_name[food_number - 1], food.food_price[food_number - 1]);
            break;
        }
        else if (yes_or_no == 2) {
            printf("그렇다면 메뉴를 다시 골라주세요.\n");
            for (int i = 0; i < 3; i++) {
                printf("%d. %s - %d원 ", i + 1, food.food_name[i], food.food_price[i]);
            }
            printf("\n");
            printf("위의 메뉴에서 하나 골라주세요 : ");
            scanf_s("%d", &food_number);
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
            continue;
        }
        else {
            printf("잘못된 입력입니다. 다시 입력해주세요.\n");
            printf("고르신 메뉴가 %s(이)가 맞습니까?\t 1. 예 2. 아니오 : ", food.food_name[food_number - 1]);
            scanf_s("%d", &yes_or_no);
        }
    } while (1);
}

```

## 운동 할 곳의 위치를 알려주는 프로그램

- 입력

1. 어떤 운동을 하고 싶은지 입력

- 출력

1. 운동을 할 수 있는 장소의 주소를 알려줌.

- 설명 : 원하는 운동 종목을 입력하면 그 장소를 알려준다.

운동 할 곳의 위치를 알려주는 프로그램을 수행하는 exercise 함수

#### - 적용된 배운 기능

운동의 종류, 운동 하는 장소, 그 장소의 주소를 저장하는 변수를 하나의 구조체

exercise\_kind로 만듦. typedef를 이용하여 Exerkind라고 새로운 별칭 지정

Initialize 함수를 이용하여 구조체 초기화

place\_name 문자열 변수를 malloc을 이용하여 동적으로 메모리 할당.

while 반복문을 사용하여 계행 문자 삭제

strcmp 함수를 사용하여 place\_name 변수와 운동의 종류 이름과 맞는지 비교하여 각각  
에 맞는 기능 실행.

free()를 통해 동적으로 할당한 메모리 삭제

```
void exercise() {
    Exerkind exercise_kind;
    initialize(&exercise_kind);
    int place_number;
    char c;
    char* place_name = (char*)malloc(sizeof(char) * strlen(exercise_kind.exercise_place[2]));
    //각 장소의 주소 저장 완료
    for (int i = 0; i < 3; i++) {
        exercise_kind.place[i] = &exercise_kind.exercise_place[i];
    }
    printf("하고자 하는 운동이 무엇인가요? : ");
    scanf_s("%s", place_name, sizeof(place_name));
    while ((c = getchar()) != '\n' && c != EOF);
    if (strcmp(place_name, "헬스") == 0) {
        printf("%s은 %x위치에 있습니다.", exercise_kind.exercise_place[0], exercise_kind.place[0]);
    }
    if (strcmp(place_name, "축구") == 0) {
        printf("%s은 %x위치에 있습니다.", exercise_kind.exercise_place[1], exercise_kind.place[1]);
    }
    if (strcmp(place_name, "테니스") == 0) {
        printf("%s는 %x위치에 있습니다.", exercise_kind.exercise_place[2], exercise_kind.place[2]);
    }
    free(place_name);
}
```

```
//4번 함수 만들기 시작
typedef struct exercise_kind {
    char type[3][10];
    char exercise_place[3][10];
    char* place[3];
}Exerkind;
```

## 구조체 멤버를 초기화 해주는 initialize 함수

### - 적용된 배운 내용

매개변수로 구조체를 포인터로 받아 사용함.

exerciseType 2차원 배열에 운동 종류로 문자열 초기화

exercisePlace 2차원 배열에 운동 하는 장소로 문자열 초기화

for문 안에서 strcpy\_s 함수를 이용하여 exercise\_kind 안의 멤버 초기화.

또한 맨 마지막에 널 문자 추가.

```
//구조체 초기화 함수
void initialize(Exerkind* exercise_kind) {
    const char* exerciseType[3] = { "헬스", "축구", "테니스" };
    const char* exercisePlace[3] = { "헬스장", "축구", "테니스" };
    for (int i = 0; i < 3; i++) {
        strcpy_s(exercise_kind->type[i], sizeof(exercise_kind->type[i]), exerciseType[i]);
        strcpy_s(exercise_kind->exercise_place[i], sizeof(exercise_kind->exercise_place[i]), exercisePlace[i]);
        exercise_kind->type[i][9] = "\0";
        exercise_kind->exercise_place[i][9] = "\0";
    }
}
```

## (2) 주차 예약 기능

### - 입력

1. 주차 자리가 필요한지 입력
2. 주차하고 싶은 층을 입력
3. 주차하고 싶은 자리를 선택

### - 출력

1. 주차 자리가 필요한지 물어보는 문장 출력
2. 몇 층에 주차를 하고 싶은지 물어보는 문장 출력
3. 자신이 원하는 층의 주차 자리 출력

4. 주차 하고 싶은 자리의 번호를 입력하게 하는 문장 출력
5. 자신이 주차한 자리는 0번으로 바뀌게 됨.

- 설명 : 주차 타워의 방식대로 몇 층 어디에 주차를 하고 싶은지를 묻고 주차를 한 곳은 0번으로 번호를 변경

주차 예약 기능을 num2\_function.h 헤더함수에 정리.

- 적용된 배운 내용

main함수 위에 너무 많은 함수들이 작성되어 수정하기 어렵다고 느껴 num2\_function.h 헤더함수를 만들고 num2\_function.c 파일에 parking\_lot 함수를 구현함.

```
1 void parking_lot(int parking_answer);
```

원하는 주차 자리를 입력 받고 주차된 자리의 번호를 바꾸는 parking\_lot 함수

- 적용된 배운 내용

parking\_lot라는 2차원 배열을 0으로 초기화 함.

이중 for문을 이용하여 parking\_lot 함수의 각 요소를 1~50번으로 다시 초기화

무한루프 while문을 이용하여 예외처리, 각 기능 수행.

If 문을 이용해서 parking\_answer의 값에 따라 기능 실행을 다르게 함.

원하는 층의 주차 자리를 출력할 때 보기 쉽게 1~9까지와 십의 자리 수 들의 위치를 맞춰주고, 주차된 자리는 0으로 만들어 줌.

각 종 예외처리는 continue문을 이용해서 처리함.

주차 자리를 고를 때 번심으로 다른 층에 주차를 하고 싶은 경우를 대비해 0번을 누르면 이전의 질문으로 돌아가도록 설정했음.

```

//주차타워 층수마다 몇대 주차 가능한지 출력하기.
//층수를 받아서 층수마다 몇 몇 층만 출력하기 ex) 1층인 경우 1층만 출력하기
//주차타워 제작 중...
void parking_lot(int parking_answer) {
    int parking_lot[5][50] = { 0 }; //여기서 다시 초기화 해서 아래에서 값을 바꿔도 초기화가 되는 바람에 결국 값이 안바뀜
    int want_number_parking_floor;
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 50; j++) {
            parking_lot[i][j] = j;
        }
    } //배열에 값 초기화 완료
    while (1)
    {
        if (parking_answer == 1) {
            printf("주차자 하고 싶은 층을 말해주세요 (1 ~ 5층 숫자만입력, 종료 : 6) : ");
            scanf_s("%d", &want_number_parking_floor);
            if (want_number_parking_floor <= 0 || want_number_parking_floor >= 7) { //예외 처리
                printf("층을 다시 입력해주세요\n\n");
                continue;
            }
            printf("\n");
            if (want_number_parking_floor == 6) {
                break;
            }
        }
        printf("%d층\n", want_number_parking_floor);
        for (int j = 0; j < 50; j++) {
            if ((j + 1) / 10 == 0) {
                printf(" ");
            }
            printf("%d ", parking_lot[want_number_parking_floor - 1][j] + 1);
            if ((j + 1) % 5 == 0) {
                printf("\n");
            }
        }
    }
}

```

```

printf("\n");
int want_number_parking_number;
printf("주차자 하고 싶은 자리를 골라주세요(0을 누르면 층을 다시 고를 수 있습니다) : "); //층 잘못골랐을 때 다시 고를 수 있게 함.
scanf_s("%d", &want_number_parking_number); //이거 index번호 아니다. -1해서 사용해야함
if (want_number_parking_number == 0) {
    continue;
}
while (1)
{
    if (want_number_parking_number < 0 || want_number_parking_number > 50) {
        printf("번호를 다시 입력해주세요.\n");
        printf("주차자 하고 싶은 자리를 골라주세요 : ");
        scanf_s("%d", &want_number_parking_number);
    }
    else {
        parking_lot[want_number_parking_floor - 1][want_number_parking_number - 1] = -1;
        break;
    }
}
}
} //주차타워 제작 완료

```

### (3) 일행 파악 기능

#### - 입력

1. 일행이 있는지 입력
2. 일행보다 먼저 왔는지 입력
3. 1번 입력 시 본인의 전화번호를 입력
4. 2번 입력 시 일행의 전화번호를 입력



- 출력

1. 일행이 있는지 물어보는 문장 출력
2. 일행보다 먼저 왔는지 물어보는 문장 출력
3. 2번 입력 시 먼저 들어온 일행이 있다면 일행에게 문자가 전송됨을 알리는 문장 출력
4. 2번 입력 시 먼저 들어온 일행이 없다면 먼저 들어온 일행이 없다는 문장 출력

- 설명 : 먼저 들어온 사람은 자신의 전화번호를 입력하여 나중에 들어오는 일행이 들어올 때 문자를 받고 나중에 들어온 사람은 일행의 전화번호를 입력하여 자신이 들어옴을 알림.

일행 파악 기능을 num3\_function.h 헤더파일에 정리함.

- 적용된 배운 내용

메인 c파일에 너무 많은 내용이 들어가 수정하기 어려워 하나의 헤더파일로 정리함. num3\_function.h 파일을 만들어 헤더파일을 만들고 num\_function.c파일에 message\_to\_party 함수를 구현함.

```
1 void message_to_party();
```

일행의 전화번호를 받고 문자를 전송하는 message\_to\_party 함수

- 적용된 배운 내용

무한루프 while문을 이용하여 예외처리 및 기능이 계속 이어질 수 있게 조정

have\_party변수를 이용하여 숫자를 받고 have\_party 변수의 값에 따라 실행되는 값이 다르게 하기 위해서 if else-if else 문을 사용함.

continue를 이용하여 예외처리를 진행했음.

```

//일행 입장 시 문자 전송 시스템
void message_to_party() {
    int have_party;
    int ask_sequence;
    int phone_number = 0;
    int party_number;
    while (1) {
        printf("일행이 있나요(1. 예 2. 아니요) : ");
        scanf_s("%d", &have_party);
        if (have_party == 1) {
            printf("일행보다 먼저왔나요? (1. 예 2. 아니요 3. 이전질문) : ");
            scanf_s("%d", &ask_sequence);
            if (ask_sequence == 1) {
                printf("본인의 전화번호를 입력해주세요( - 제외하고 입력) : ");
                scanf_s("%d", &phone_number); //int라서 010이 10으로 저장됨
            }
            else if (ask_sequence == 2) {
                printf("일행의 전화번호를 입력해주세요( - 제외하고 입력) : ");
                scanf_s("%d", &party_number);
                if (party_number == phone_number) {
                    printf("일행에게 문자가 전송됐습니다.\n");
                    break;
                }
                else if (party_number != phone_number) {
                    printf("먼저 들어온 일행이 없습니다.\n");
                }
            }
            else if (ask_sequence == 3) {
                continue;
            }
            else {
                printf("번호를 다시 입력해주세요.");
            }
        }
        else if (have_party == 2) { //2번 입력시 종료
            break;
        }
        else { //예외처리
            printf("번호를 다시 입력해주세요.\n");
        }
    }
}

```

## 4. 테스트 결과

### (1) 고객이 이곳에 온 목적을 파악하고 이를 도와주는 기능

#### 1. 영화 예매 기능

- 보고 싶은 영화의 장르를 입력하면 그에 맞는 영화 제목 출력 후 영화 선택 시 그 영화의 시간표 출력 후 원하는 시간 번호 입력 시 제대로 골랐는지 확인 후 "예"가 나올 경우 예매 완료

- 테스트 결과 스크린샷

보고싶은 장르를 골라주세요

1. 애니메이션
2. 히어로
3. 액션
4. 종료 : 1

무엇을 보고 싶으신가요?

1. 엘리멘탈
2. 겨울왕국3
3. 미니언즈 : 1

1. 10시20분
2. 11시00분
3. 11시40분
4. 12시40분
5. 13시30분
6. 14시20분
7. 15시20분
8. 16시10분
9. 17시00분

위의 시간표에서 시간을 골라주세요(숫자 작성) : 1

10시20분이 선택한 시간이 맞습니까? (예, 아니오로 대답해주세요) : 1

답변을 다시 선택해주세요.

10시20분이 선택한 시간이 맞습니까? (예, 아니오로 대답해주세요) : dksldh

답변을 다시 선택해주세요.

10시20분이 선택한 시간이 맞습니까? (예, 아니오로 대답해주세요) : 아니오

그렇다면 원하는 시간을 다시 골라주세요.

위의 시간표에서 시간을 골라주세요(숫자 작성) : 2

11시00분이 선택한 시간이 맞습니까? (예, 아니오로 대답해주세요) : 예

11시00분으로 예매 완료 되었습니다. 시작 10분전에 입장해주세요.

## 2. 쇼핑 할 곳의 위치 정보 제공 기능

- 쇼핑하고자 하는 품목을 선택하면 그에 맞는 기능들이 시작 됨. 의류, 명품의

경우는 브랜드를 선택하고 이후 그 브랜드의 위치를 알려줌. 4번을 누르면 종료

- 테스트 결과 스크린샷

이곳에 온 목적이 무엇입니까?  
 1. 영화관 2. 쇼핑(식자재, 의류) 3. 식당 4. 운동(헬스, 축구, 테니스) 5. 종료 : 2  
 쇼핑할 품목이 무엇인가요?  
 1. 식자재 2. 의류 3. 명품 4. 종료 : 1  
 지하 1층에 이마트가 있습니다.

쇼핑할 품목이 무엇인가요?  
 1. 식자재 2. 의류 3. 명품 4. 종료 : 2  
 어떤 브랜드를 원하십니까?  
 1. 스파이더 2. 나이키 3. 아디다스 4. MLB 5. 8seconds 6. 유니클로 7. FILA 8. 무신사 9. 스투시 : 5  
 1. 스파이더 2. 나이키 3. 아디다스  
 4. MLB 5. 8seconds 6. 유니클로  
 7. FILA 8. 무신사 9. 스투시  
 8seconds는 위의 표의 5번에 있습니다.

쇼핑할 품목이 무엇인가요?  
 1. 식자재 2. 의류 3. 명품 4. 종료 : 3  
 어떤 브랜드를 원하십니까?  
 1. CHANEL 2. DIOR 3. BALENCIAGA 4. GUCCI 5. Yves Saint Laurent 6. PRADA 7. Louis Vuitton 8. Hermes 9. Burberry : 5  
 CHANEL DIOR BALENCIAGA  
 GUCCI Yves Saint Laurent PRADA  
 Louis Vuitton Hermes Burberry  
 Yves Saint Laurent는 위의 표의 5번에 있습니다.

쇼핑할 품목이 무엇인가요?  
 1. 식자재 2. 의류 3. 명품 4. 종료 : 4

### 3. 식당에서의 메뉴 주문 프로그램

- 원하는 종류의 음식을 고르면 그에 맞는 메뉴가 출력됨. 메뉴를 고르면 고른 메뉴가 맞는지 확인하고 1번을 입력하면 메뉴와 가격을 출력, 2번을 입력하면 다시 메뉴를 고르게 유도, 그 이외의 답을 입력하면 메뉴가 맞는지를 다시 입력하게 함.

- 테스트 결과 스크린샷

어떤 종류의 음식을 드시고 싶으신가요?  
 1. 한식 2. 일식 3. 중식 : 1  
 1. 제육볶음 - 7000원 2. 김치찌개 - 6000원 3. 돌솥비빔밥 - 6000원  
 위의 메뉴에서 하나 골라주세요 : 1  
 고르신 메뉴가 제육볶음(이)가 맞습니까? 1. 예 2. 아니오 : 3  
 잘못된 입력입니다. 다시 입력해주세요.  
 고르신 메뉴가 제육볶음(이)가 맞습니까? 1. 예 2. 아니오 : 2  
 그렇다면 메뉴를 다시 골라주세요.  
 1. 제육볶음 - 7000원 2. 김치찌개 - 6000원 3. 돌솥비빔밥 - 6000원  
 위의 메뉴에서 하나 골라주세요 : 2  
 고르신 메뉴가 김치찌개(이)가 맞습니까? 1. 예 2. 아니오 : 1  
 김치찌개 6000원입니다.

### 4. 운동 할 곳의 위치를 알려주는 프로그램

- 원하는 운동의 종류를 입력 시 운동을 할 수 있는 장소의 주소를 알려줌.
- 테스트 결과 스크린샷

하고자 하는 운동이 무엇인가요? : 헬스  
헬스장은 348ff88e 위치에 있습니다.

## (2) 주차 예약 기능

- 주차 할 자리가 필요한지 물어본 것에 대한 대답이 1번일 경우 주차하기를 원하는 층을 입력하고 그 층에서 주차 할 수 있는 자리를 보여주게 된다. 주차하고 싶은 자리를 입력하게 하고 고른 자리는 0으로 표시되어 주차 할 수 없게 된다.

- 테스트 결과 스크린샷

```
주차 자리가 필요하십니까? 1. 예 2. 아니오 : 1
주차할 하고 싶은 층을 말해주세요 (1 ~ 5층 숫자만입력, 종료 : 6) : 3

3층
 1  2  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
26 27 28 29 30
31 32 33 34 35
36 37 38 39 40
41 42 43 44 45
46 47 48 49 50

주차 하고 싶은 자리를 골라주세요(0을 누르면 층을 다시 고를 수 있습니다) : 2
주차할 하고 싶은 층을 말해주세요 (1 ~ 5층 숫자만입력, 종료 : 6) : 3

3층
 1  0  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
26 27 28 29 30
31 32 33 34 35
36 37 38 39 40
41 42 43 44 45
46 47 48 49 50
```

주차 하고 싶은 자리를 골라주세요(0을 누르면 층을 다시 고를 수 있습니다) : 0  
주차 를 하고 싶은 층을 말해주세요 (1 ~ 5층 숫자만 입력, 종료 : 6) : 5

5층

1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25  
26 27 28 29 30  
31 32 33 34 35  
36 37 38 39 40  
41 42 43 44 45  
46 47 48 49 50

주차 하고 싶은 자리를 골라주세요(0을 누르면 층을 다시 고를 수 있습니다) : 3  
주차 를 하고 싶은 층을 말해주세요 (1 ~ 5층 숫자만 입력, 종료 : 6) : 5

5층

1 2 0 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25  
26 27 28 29 30  
31 32 33 34 35  
36 37 38 39 40  
41 42 43 44 45  
46 47 48 49 50

### (3) 일행 파악 기능

- 일행이 있는지를 물어보고 자신이 일행보다 먼저 도착했는지를 물어본 후에 먼저 도착했다고 하면 자신의 전화번호를 입력하게 하고 일행보다 나중에 도착했다면 일행의 전화번호를 입력하게 만든다. 일행이 있다면 먼저 들어온 사람에게는 자신의 일행이 들어왔다고 문자가 오게 된다.

- 테스트 결과 스크린샷

일행이 있나요(1. 예 2. 아니요) : 1  
일행보다 먼저왔나요? (1. 예 2. 아니요 3. 이전 질문) : 1  
본인의 전화번호를 입력해주세요( - 제외하고 입력) : 01024821003  
일행이 있나요(1. 예 2. 아니요) : 1  
일행보다 먼저왔나요? (1. 예 2. 아니요 3. 이전 질문) : 2  
일행의 전화번호를 입력해주세요( - 제외하고 입력) : 01024821003  
일행에게 문자가 전송되었습니다.

## 5. 계획 대비 변경 사항

### 1) 기능 이름 변경

- 이전 : 기능 1, 기능 2, 기능 3
- 이후 : 기능 1 -> 고객이 이곳에 온 목적을 파악하고 이를 도와주는 기능  
기능 2 -> 주차 예약 기능  
기능 3 -> 일행 파악 기능
- 사유 : 기능 1, 기능 2, 기능 3의 이름을 입력하지 않아서 기능의 이름을 만들게 됨.

### 2) 일행 파악 기능 일부 축소

- 이전 : 일행의 위치까지 파악하도록 설정.
- 이후 : 위치 파악 기능 삭제
- 사유 : 위치 파악 기능을 만들기 위해서는 1번의 기능들과 모두 연동 시켜야 하는데, 이 부분을 구현하는데 한계점이 있다고 파악했음.

## 6. 느낀점

2학기 초반 c 수업을 듣기 위해 OT를 들을 때 깃허브를 사용한다고 했을 때 막막했고, 이걸 어떻게 하나 싶었지만 계속 쓰다 보니 오히려 이젠 없으면 허전하다. 그리고 확실히 필요한 내용만 짧고 굵게 많이 알아 간 것 같다. 처음엔 수업 자체가 빠세고 어렵다고 생각했는데, 이미 알고 있던 내용 외에도 알게 된 것이 많아 좋았다. 교수님도 매우 열정적으로 수업하시고 질문에도 답을 잘해주셔서 좋았고 얻어가는 것이 많은 수업이었다.