

C프로그래밍및실습

대중교통 예매시스템

최종 보고서

제출일자: 2023/12/23

제출자명: 김영재

제출자학번: 230114

1. 프로젝트 목표

1) 배경 및 필요성

현대 사회에서는 빠르게 이동하고 편리하게 교통수단을 이용하는 요구가 점차 증가함. 전통적인 예매 및 승차 시스템은 불편한 예매 프로세스와 긴 대기시간 등으로 고객의 교통 서비스의 만족도를 저하시키고 수익 감소를 야기할 수 있음. 때문에 문제를 해결하기 위해선 현대적이고 효율적인 대중 교통 예매 시스템이 필요함.

2) 프로젝트 목표

고객들이 원하는 목적지, 교통수단, 날짜, 좌석을 예매하고, 결제 내역 및 스케줄 확인을 보여 주는 프로그램을 만드는 것을 목표로 함.

3) 차별점

기존 프로그램들은 각 교통수단마다 예매 프로그램을 이용해 고객의 불편함을 야기할 수 있음. 이 프로그램은 다양한 교통 수단을 통합하여 예매하고 통합된 승차 티켓을 제공하는 것에 기존 프로그램과 차별점이 있음.

2. 기능 계획

1) 기능 1 (메인 기능 구현)

- 설명 : 고객의 요구에 따라 원하는 프로그램을 작동하는 기능

(1) 세부 기능 1 (고객 거주지 입력 및 변경)

- 설명 : 예매를 하기 위해 고객의 거주지를 입력 받거나 요구에 따라 변경한다.

(2) 세부 기능 2 (예매하기)

- 설명 : 고객이 희망하는 목적지와 교통수단을 입력 받고 가격을 계산해준다.

(3) 세부 기능 3 (예매 취소하기)

- 설명 : 예매했던 정보들 중 고객이 취소 하고싶은 예매 정보를 취소한다.

(4) 세부 기능 4 (결제 내역 및 예매 정보 확인)

- 설명 : 고객의 과거 예매 정보와 결제 내역을 보여준다.

(5) 세부 기능 5 (프로그램 종료)

- 설명 : 프로그램을 종료 시킨다

2) 기능 2 (대중 교통 예매)

- 설명 : 고객의 거주지에 맞춰 대중 교통을 예매하는 기능

(1) 세부 기능 1 (목적지 입력 받기)

- 설명 : 고객이 원하는 목적지(광역시만)를 입력한다.

(2) 세부 기능 2 (날짜 입력 받기)

- 설명 : 고객이 원하는 날짜를 입력 받는다.

(3) 세부 기능 3 (교통 수단 입력 받기)

- 설명 : 거주지와 목적지가 같을 경우, 택시와 지하철 중 고객이 원하는 교통수단을 입력 받고 다를 경우, 버스와 기차 중 원하는 교통수단을 입력 받는다

3. 기능 구현

(1) 메인 메뉴

- 입출력 : 입력 - 1, 2, 3, 4,5 / 출력 - 1. 거주지 변경 2. 예매하기 3.예매 취소

4. 결제 내역 및 예매 정보 확인 5. 종료

- 설명 : 사용자로부터 메인 메뉴의 기능들 중에서 번호를 입력 받고 입력받은 번호의 기능을 실행한다.

- 적용된 배운 내용 : scanf_s 함수, while 반복문, switch 조건문
- 코드 스크린샷

```
int main() {
    int reservationCount = 0; // 예약된 횟수를 저장하는 변수
    Reservation reservations[MAX_RESERVATIONS]; // 여러 예약 정보를 저장하기 위한 배열

    enterLocation(&reservations[reservationCount]);

    while (1) {
        int menu = -1; // 입력한 메뉴 번호를 저장하는 변수

        // 메인 메뉴 선택
        printf("-----\n");
        printf("메뉴를 입력해주세요\n");
        printf("1. 출발지 변경\n2. 예약하기\n3. 예약 취소\n4. 결제 내역 및 예약 정보 확인\n5. 종료\n");
        printf("-----\n");
        scanf_s("%d", &menu);

        int terminate = 0; // 프로그램 종료를 위한 변수

        switch (menu) {
            case 1: // 1. 출발지 변경
                changeLocation(&reservations[reservationCount]);
                break;
            case 2: // 2. 예약하기
                makeReservation(reservations, &reservationCount);
                break;
            case 3: // 3. 예약 취소
                cancelReservation(reservations, &reservationCount);
                break;
            case 4: // 4. 결제 내역 및 예약 정보 확인
                showReservationInfo(reservations, reservationCount);
                break;
            case 5: // 5. 프로그램 종료
                terminate = 1;
                break;
            default:
                printf("잘못 입력하셨습니다. 다시 선택해주세요.\n");
        }

        if (terminate == 1) {
            printf("프로그램을 종료합니다.");
            break;
        }
    }
}
```

(2) 출발지 입력 및 변경

- 입출력 : 입력 - 1,2,3,4,5,6

출력 - "고객님의 출발지는 %s입니다.", "고객님의 출발지가 %s(으)로 변경되었습니다."

- 설명 : 사용자로부터 출발지를 입력 받고 변경한다.
- 적용된 배운 내용 : scanf_s 함수, switch문, while문, 구조체 포인터,
- 코드 스크린샷

<pre>// 출발지 입력 함수 void enterLocation(Reservation* reservation) { printf("대중교통 예약 프로그램을 실행합니다.\n"); while (1) { printf("고객님의 출발지를 입력해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)\n"); scanf_s("%d", &reservation->number); switch (reservation->number) { case 1: strcpy_s(reservation->location, sizeof(reservation->location), "부산"); break; case 2: strcpy_s(reservation->location, sizeof(reservation->location), "대구"); break; case 3: strcpy_s(reservation->location, sizeof(reservation->location), "대전"); break; case 4: strcpy_s(reservation->location, sizeof(reservation->location), "광주"); break; case 5: strcpy_s(reservation->location, sizeof(reservation->location), "인천"); break; case 6: strcpy_s(reservation->location, sizeof(reservation->location), "울산"); break; default: printf("유효하지 않은 번호입니다. 다시 입력해주세요.\n"); continue; // 올바른 번호를 입력할 때까지 반복 } printf("고객님의 출발지는 %s입니다.\n", reservation->location); break; // 올바른 번호를 입력하면 반복문 종료 } }</pre>	<pre>// 출발지 변경 함수 void changeLocation(Reservation* reservation) { printf("변경하고 싶은 고객님의 출발지를 입력해주세요.(1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)\n"); scanf_s("%d", &reservation->number); switch (reservation->number) { case 1: strcpy_s(reservation->location, sizeof(reservation->location), "부산"); break; case 2: strcpy_s(reservation->location, sizeof(reservation->location), "대구"); break; case 3: strcpy_s(reservation->location, sizeof(reservation->location), "대전"); break; case 4: strcpy_s(reservation->location, sizeof(reservation->location), "광주"); break; case 5: strcpy_s(reservation->location, sizeof(reservation->location), "인천"); break; case 6: strcpy_s(reservation->location, sizeof(reservation->location), "울산"); break; default: printf("유효하지 않은 번호입니다. 출발지 변경에 실패했습니다.\n"); return; // 함수 종료 } printf("고객님의 출발지가 %s(으)로 변경되었습니다.\n", reservation->location); }</pre>
--	--

(3) 예매하기

- 입출력 : 입력 - 목적지, 교통수단, 거리 ,날짜

출력 - "%s로 %dm 이동하여 %d원 결제되었습니다.",

"%d원 결제되었습니다."

- 설명 : 사용자로부터 목적지를 입력 받은 후 거주지와 목적지가 같을 경우 교통수단 및 거리를 입력 받고, 거주지와 목적지가 다를 경우 날짜 및 교통수단을 입력 받는다.

- 적용된 배운 내용 : 구조체, scanf_s 함수, 조건문, while문, do-while문, strcmp 함수, 포인터, switch문, strcpy_s 함수, 배열, 비교 연산자, 산술 연산자, 증감 연산자

- 코드 스크린샷

```
0. 예매하기 함수
void makeReservation(Reservation reservations[], int* reservationCount) {
    if (*reservationCount == MAX_RESERVATIONS) {
        printf("예약 가능 가능 한도로 다 찬성으로 프로그램을 종료합니다.\n");
        return; // 프로그램 종료
    }
    // 예약 정보를 저장할 구조체의 인덱스
    int currentIndex = *reservationCount;
    while (1) {
        // 예약을 받은 입력
        printf("예약하시는 목적지를 선택해주세요 (0: 1. 부산 2. 대구 3. 대전 4. 광주 5. 인천 6. 울산 등):\n");
        scanf_s("%d", &reservations[currentIndex].number);
        // 예약을 받은 목적지 번호
        switch (reservations[currentIndex].number) {
            case 1:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "부산");
                break;
            case 2:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "대구");
                break;
            case 3:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "대전");
                break;
            case 4:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "광주");
                break;
            case 5:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "인천");
                break;
            case 6:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "울산");
                break;
            default:
                printf("목적지 번호가 잘못되었습니다. 다시 입력해주세요.\n");
                // 올바른 번호를 입력할 때까지 반복
                continue;
        }
        // 거주지와 목적지가 같은지 확인
        if (strcmp(reservations[currentIndex].location, reservations[currentIndex].destination) == 0) {
            // 교통수단 입력
            do {
                printf("예약하시는 교통수단을 선택해주세요 (0: 1. 지하철 2. 버스):\n");
                scanf_s("%d", &reservations[currentIndex].transportation, (&reservations[currentIndex].transportation));
                // 1. 지하철을 입력하지 않았을 경우
                if (strcmp(reservations[currentIndex].transportation, "1") != 0 && strcmp(reservations[currentIndex].transportation, "2") != 0) {
                    printf("목적지 번호가 잘못되었습니다. 다시 입력해주세요.\n");
                }
            } while (strcmp(reservations[currentIndex].transportation, "1") != 0 && strcmp(reservations[currentIndex].transportation, "2") != 0);
            // 거리 입력
            printf("예약하시는 목적지까지 걸리는 거리를 입력해주세요 (0 단위로):\n");
            scanf_s("%d", &reservations[currentIndex].distance);
            // 교통수단 번호가 잘못되었는지 확인
            if (strcmp(reservations[currentIndex].transportation, "1") == 0) {
                reservations[currentIndex].price = 4300 + ((reservations[currentIndex].distance / 150) * 100);
            } else if (strcmp(reservations[currentIndex].transportation, "2") == 0) {
                reservations[currentIndex].price = 3000 + ((reservations[currentIndex].distance / 100) * 100);
            }
        }
    }
}
```

```
1. 예약 취소 함수
void cancelReservation(Reservation reservations[], int* reservationCount) {
    if (*reservationCount == 0) {
        printf("예약 취소할 예약이 없습니다.\n");
        return;
    }
    // 예약을 취소할 인덱스
    int currentIndex = *reservationCount;
    while (1) {
        // 예약을 취소할 인덱스
        printf("예약하시는 목적지를 선택해주세요 (0: 1. 부산 2. 대구 3. 대전 4. 광주 5. 인천 6. 울산 등):\n");
        scanf_s("%d", &reservations[currentIndex].number);
        // 예약을 취소할 목적지 번호
        switch (reservations[currentIndex].number) {
            case 1:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "부산");
                break;
            case 2:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "대구");
                break;
            case 3:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "대전");
                break;
            case 4:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "광주");
                break;
            case 5:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "인천");
                break;
            case 6:
                strcpy_s(reservations[currentIndex].destination, sizeof(reservations[currentIndex].destination), "울산");
                break;
            default:
                printf("목적지 번호가 잘못되었습니다. 다시 입력해주세요.\n");
                continue;
        }
        // 거주지와 목적지가 같은지 확인
        if (strcmp(reservations[currentIndex].location, reservations[currentIndex].destination) == 0) {
            // 교통수단 입력
            do {
                printf("예약하시는 교통수단을 선택해주세요 (0: 1. 지하철 2. 버스):\n");
                scanf_s("%d", &reservations[currentIndex].transportation, (&reservations[currentIndex].transportation));
                // 1. 지하철을 입력하지 않았을 경우
                if (strcmp(reservations[currentIndex].transportation, "1") != 0 && strcmp(reservations[currentIndex].transportation, "2") != 0) {
                    printf("목적지 번호가 잘못되었습니다. 다시 입력해주세요.\n");
                }
            } while (strcmp(reservations[currentIndex].transportation, "1") != 0 && strcmp(reservations[currentIndex].transportation, "2") != 0);
            // 거리 입력
            printf("예약하시는 목적지까지 걸리는 거리를 입력해주세요 (0 단위로):\n");
            scanf_s("%d", &reservations[currentIndex].distance);
            // 교통수단 번호가 잘못되었는지 확인
            if (strcmp(reservations[currentIndex].transportation, "1") == 0) {
                reservations[currentIndex].price = 4300 + ((reservations[currentIndex].distance / 150) * 100);
            } else if (strcmp(reservations[currentIndex].transportation, "2") == 0) {
                reservations[currentIndex].price = 3000 + ((reservations[currentIndex].distance / 100) * 100);
            }
        }
    }
}
```

(4) 예매취소

- 입출력 : 입력 - cancelIndex

출력 - "예매가 취소되었습니다."

- 설명 : 사용자가 예매한 정보들 중 취소 하고싶은 예매 정보를 취소한다.
- 적용된 배운 내용 : scanf_s문, if문, for문, 포인터, 구조체, strcpy_s함수, 증감 연산자, 비교 연산자
- 코드 스크린샷

```
// 예매 취소 함수
void cancelReservation(Reservation reservations[], int* reservationCount) {
    int cancelIndex;
    printf("취소할 예매 번호를 입력하세요: ");
    scanf_s("%d", &cancelIndex);
    if (*reservationCount == 0) {
        printf("예매한 정보가 없습니다.\n");
    }
    else {
        // 예매 번호가 유효한 범위 내에 있는지 확인
        if (cancelIndex >= 1 && cancelIndex <= *reservationCount) {
            // 선택한 예매 정보를 삭제하고 나머지 정보를 쉬프트
            for (int i = cancelIndex - 1; i < *reservationCount - 1; i++) {
                strcpy_s(reservations[i].location, sizeof(reservations[i].location), reservations[i + 1].location);
                strcpy_s(reservations[i].transportation, sizeof(reservations[i].transportation), reservations[i + 1].transportation);
                reservations[i].date[0] = reservations[i + 1].date[0];
                reservations[i].date[1] = reservations[i + 1].date[1];
                reservations[i].date[2] = reservations[i + 1].date[2];
                reservations[i].distance = reservations[i + 1].distance;
                reservations[i].price = reservations[i + 1].price;
            }
            (*reservationCount)--;
            printf("예매가 취소되었습니다.\n");
        }
        else {
            printf("유효하지 않은 예매 번호입니다.\n");
        }
    }
}
```

(5) 결제 내역 및 예매 정보 확인

- 입출력 : 입력 - menu에 3 입력

출력 - "예매한 정보를 확인합니다.

----- 예매 정보 -----

예매 번호: %d

교통수단 : %s

이동 거리: %dm

결제 금액 : %d원

-----",

"----- 예매 정보 -----

예매 번호 : %d

거주지 : %s

목적지 : %s

이동 날짜: %d년 %d월 %d일

교통수단 : %s

결제 금액: %d원

-----"

- 설명 : 메인 메뉴 선택에서 3번 선택 시 예매한 정보를 출력해준다.
- 적용된 배운 내용 : switch문, 조건문, for문, 구조체, 배열
- 코드 스크린샷

```
// 예매 정보 확인 함수
void showReservationInfo(Reservation reservations[], int reservationCount) {
    printf("예매한 정보를 확인합니다.\n");
    if (reservationCount > 0) {
        for (int i = 0; i < reservationCount; i++) {
            if (strcmp(reservations[i].location, reservations[i].destination) == 0) {
                printf("----- 예매 정보 ----- \n");
                printf("예매 번호: %d\n", i + 1);
                printf("출발지: %s\n", reservations[i].location);
                printf("교통수단: %s\n", reservations[i].transportation);
                printf("이동 거리: %d\n", reservations[i].distance);
                printf("결제 금액: %d원\n", reservations[i].price);
                printf("----- \n");
            }
            else {
                printf("----- 예매 정보 ----- \n");
                printf("예매 번호: %d\n", i + 1);
                printf("출발지: %s\n", reservations[i].location);
                printf("목적지: %s\n", reservations[i].destination);
                printf("이동 날짜: %d년 %d월 %d일\n", reservations[i].date[0], reservations[i].date[1], reservations[i].date[2]);
                printf("교통수단: %s\n", reservations[i].transportation);
                printf("결제 금액: %d원\n", reservations[i].price);
                printf("----- \n");
            }
        }
    }
    else {
        printf("예매된 정보가 없습니다.\n");
    }
}
```

(6) 프로그램 종료

- 입출력 : 입력 – menu에 5 입력

출력 – “프로그램을 종료합니다.”

- 설명 : 프로그램을 종료 시킨다.
- 적용된 배운 내용: while문, switch문, if문

- 코드 스크린샷

```
case 5: // 5. 프로그램 종료
    terminate = 1;
    break;
}
if (terminate == 1) {
    printf("프로그램을 종료합니다.");
    break;
}
```

4. 테스트 결과

(1) 메인 메뉴 에러

- 설명 : 사용자가 메인 메뉴의 1번부터 5번까지의 번호를 제외한 번호를 입력하였을 때 오류 문구와 함께 다시 입력 받는다.

- 테스트 결과 스크린샷

```
-----
메뉴를 입력해주세요
1. 출발지 변경
2. 예매하기
3. 예매 취소
4. 결제 내역 및 예매 정보 확인
5. 종료
-----
6
잘못 입력하셨습니다. 다시 선택해주세요.
```

(2) 출발지 입력 및 예외 처리

- 설명 : 프로그램 실행 시 사용자로부터 번호를 입력 받아 출발지를 저장하고, 범위 내의 번호를 입력하지 않을 시 오류 메시지와 함께 다시 입력 받는다.

- 테스트 결과 스크린샷

```
대중교통 예매 프로그램을 실행합니다.
고객님의 출발지를 입력해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)
7
유효하지 않은 번호입니다. 다시 입력해주세요.
고객님의 출발지를 입력해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)
1
고객님의 출발지는 부산입니다.
```

(3) 출발지 변경 및 예외 처리

- 설명 : 사용자로부터 번호를 입력 받아 원하는 출발지로 변경 하고 범위 내의 번호를 입력하지 않을 시 오류 메시지와 함께 기능을 종료한다.

- 테스트 결과 스크린샷

<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 1 변경하고 싶은 고객님의 출발지를 입력해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산) 7 유효하지 않은 번호입니다. 출발지 변경에 실패했습니다. </pre>	<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 1 변경하고 싶은 고객님의 출발지를 입력해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산) 2 고객님의 출발지가 대구(으)로 변경되었습니다. </pre>
---	--

(4) 예매 하기 (출발지 = 목적지, 출발지 != 목적지) 및 예외 처리

- 설명 : 사용자로부터 목적지를 입력 받은 후 출발지와 일치할 경우 교통수단과 거리를 입력 받고, 출발지와 일치하지 않을 경우 날짜와 거리를 입력 받은 후 계산해준다. 그리고 범위 내의 목적지 번호를 입력 하지 않을 시 오류 메시지와 함께 다시 입력 받는다.

- 테스트 결과 스크린샷

```

-----
메뉴를 입력해주세요
1. 출발지 변경
2. 예매하기
3. 예매 취소
4. 결제 내역 및 예매 정보 확인
5. 종료
-----
2
원하시는 목적지를 선택해주세요.
(1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)
7
유효하지 않은 번호입니다. 다시 입력해주세요.
원하시는 목적지를 선택해주세요.
(1.부산 2.대구 3.대전 4.광주 5.인천 6.울산)

```

<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 2 원하시는 목적지를 선택해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산) 1 원하시는 교통수단을 선택해주세요.(택시, 지하철) 택시 원하는 목적지까지 걸리는 거리를 입력해주세요.(m 단위로) 5000 택시로 5000m 이동하여 7600원 결제되었습니다. </pre>	<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 2 원하시는 목적지를 선택해주세요. (1.부산 2.대구 3.대전 4.광주 5.인천 6.울산) 2 이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분) 2023 12 23 원하시는 교통수단을 선택해주세요.(고속버스, 기차) 고속버스 20000원 결제되었습니다. </pre>
---	--

(5) 예매 교통수단 예외 처리

- 설명 : 사용자가 지정된 교통수단을 입력 하지 않을 시 오류 메시지와 함께 다시 입력 받는다.

- 테스트 결과 스크린샷

원하시는 교통수단을 선택해주세요.(택시, 지하철)
버스
유효하지 않은 교통수단입니다. 다시 입력해주세요.
원하시는 교통수단을 선택해주세요.(택시, 지하철)

원하시는 교통수단을 선택해주세요.(고속버스, 기차)
택시
유효하지 않은 교통수단입니다. 다시 입력해주세요.
원하시는 교통수단을 선택해주세요.(고속버스, 기차)

(6) 예매 날짜 예외 처리

- 설명 : 사용자가 2023~2024년을 벗어나거나 유효하지 않은 월, 일자를 입력 할 시 오류 메시지와 함께 다시 입력 받는다.

- 테스트 결과 스크린샷

이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)
2025 1 1
유효하지 않은 년도입니다. 2023년~2024년 사이 날짜를 입력해주세요.

이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)
2023 13 1
유효하지 않은 월입니다. 1월부터 12월 사이의 값을 입력해주세요.

2023 2 29
유효하지 않은 일자입니다. 1일부터 28일 사이의 값을 입력해주세요.
이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)
2023 4 31
유효하지 않은 일자입니다. 1일부터 30일 사이의 값을 입력해주세요.
이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)

(6) 결제 내역 및 예매 정보 확인

- 설명 : 저장 되어있는 예매 정보를 출력한다.

- 테스트 결과 스크린샷

<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 4 예매한 정보를 확인합니다. ----- 예매 정보 ----- 예매 번호: 1 출발지: 부산 교통수단: 택시 이동 거리: 5000m 결제 금액: 7600원 ----- ----- 예매 정보 ----- 예매 번호: 2 출발지: 부산 목적지: 대구 이동 날짜: 2023년 12월 23일 교통수단: 기차 결제 금액: 50000원 ----- ----- 예매 정보 ----- 예매 번호: 3 출발지: 대전 목적지: 광주 이동 날짜: 2024년 2월 27일 교통수단: 기차 결제 금액: 50000원 ----- </pre>	<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 4 예매한 정보를 확인합니다. 예매된 정보가 없습니다. </pre>
--	---

(7) 예매 취소 및 예외 처리

- 설명: 사용자가 예매한 정보들 중 취소 하고싶은 예매 정보를 취소하고 예매 번호들을 앞당긴다. 그리고 취소할 예매 정보가 없거나 유효하지 않은 예매 번호를 입력할 시

오류 메시지와 함께 기능을 종료한다.

- 테스트 결과 스크린샷

<pre> ----- 메뉴를 입력해주세요 1. 출발지 변경 2. 예매하기 3. 예매 취소 4. 결제 내역 및 예매 정보 확인 5. 종료 ----- 3 취소할 예매 번호를 입력하세요: 2 예매가 취소되었습니다. </pre>	<pre> ----- 예매 정보 ----- 예매 번호: 1 출발지: 부산 교통수단: 택시 이동 거리: 5000m 결제 금액: 7600원 ----- ----- 예매 정보 ----- 예매 번호: 2 출발지: 대전 목적지: 광주 이동 날짜: 2024년 2월 27일 교통수단: 기차 결제 금액: 50000원 ----- </pre>
---	--

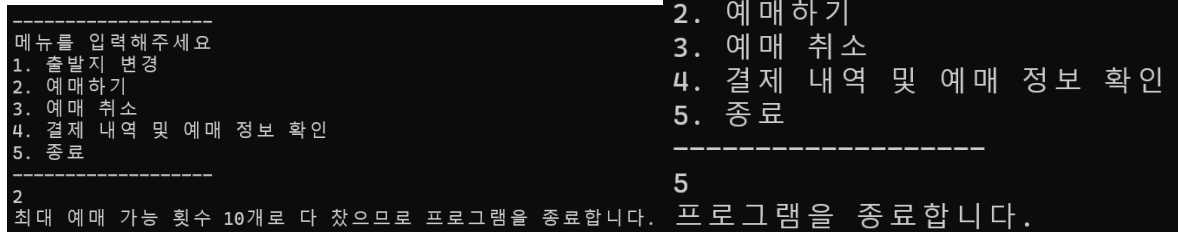
취소할 예매 번호를 입력하세요: 4 유효하지 않은 예매 번호입니다.	취소할 예매 번호를 입력하세요: 1 예매한 정보가 없습니다.
--	--------------------------------------

(8) 프로그램 종료

- 설명 : 예매를 10번 하거나 메인 메뉴의 5가지 기능 중 5번째 기능 실행 시

프로그램을 종료 한다.

- 테스트 결과 스크린샷



5. 계획 대비 변경 사항

1) 가격 계산

- 이전 : 거주지와 목적지가 같을 때 사용자로부터 목적지까지 걸리는 시간을 입력 받고 원하는 교통수단에 따라 시간에 관한 계산식을 만듦
- 이후 : 사용자로부터 목적지까지 걸리는 시간이 아닌 거리를 입력 받고 원하는 교통수단에 따라 거리에 관한 계산식을 만듦
- 사유 : 시간에 관한 계산식보다 거리에 관한 계산식이 더욱 구체적이고 현실적이기 때문에

2) 예매 제한

- 이전 : 예매 횟수에 제한을 두지 않았다.
- 이후 : 최대 예매 횟수를 10개로 두고 10번을 한 경우 오류 메시지와 함께 프로그램을 종료 한다.
- 사유 : 데이터 크기 제한과 프로그램 제한을 둘 필요가 있어서

3) 예매 취소

- 이전 : 예매 취소 기능이 없었음
- 이후 : 사용자가 예매한 정보들 중 취소 하고싶은 예매 정보를 취소하는 기능을 만듦

- 사유 : 예매 제한이 생겼으므로 예매 정보를 저장하는 것 뿐만 아닌 취소하는 기능도 필요하다는 것을 느껴서

4) 출발지, 목적지 번호로 입력 받기

- 이전 : 출발지, 목적지를 지역명으로 입력 받음
- 이후 : 출발지, 목적지를 번호로 입력 받아 해당 번호의 지역명을 저장함
- 사유 : 지정되지 않은 지역을 입력 받지 않도록 할 때 지역명으로 설정하면 너무 복잡하고 코드가 길어지므로 번호를 통해 입력 받아 코드를 간단하고 효율적으로 바꿈

6. 느낀점

한 학기 동안 C프로그래밍및실습 강의를 듣고 나서, 프로그래밍과 컴퓨터과학에 대한 새로운 이해와 경험을 하였습니다. 1학기때 배운 python에선 배우지 못했던 구조체, 포인터, 헤더파일 등과 반복문, 조건문, 함수 선언 등의 개념을 익히면서 언어 별 차이점을 느끼고 프로그램이 어떻게 동작하는지를 깨달을 수 있었습니다. 또한, 프로젝트를 통해 현실적인 문제를 해결하는 능력을 키우고 코드를 디버깅하고 최적화하는 과정에서 더 많은 것을 배울 수 있었습니다. 특히, 예기치 못한 오류에 대처하고 문제를 해결하는 과정에서 문제 해결 능력이 크게 향상된 것 같습니다.