

**사회학과**

**201721899 송지민**

**나만의 티켓 예매 프로그램 작성하기**

* 공연제목 선택
* 성인/ 청소년 티켓 수량 선택
* 장소 및 공연시간 선택
* 좌석 예약
* 결제금액 계산
* 최종 결과 및 결제금액 출력

**<프로그래밍 언어>**

**중간고사 대체 프로그래밍 과제**

1. **프로그램 설명**
2. **코드 및 주석**
3. **정상 실행화면**
4. **오류 발생시 실행화면**
5. **보고서를 마치며**

**목차**

1. **프로그램 설명**

본 프로그램은 극장 예매 프로그램으로, <프로그래밍 언어> 중간고사 대체 과제의 일환으로 작성하였다.

프로그램의 기본 기능은 지시 사항에 따라 구현하였으며, 그 대략적인 내용은 다음과 같다.

* 사용자로부터 공연 제목 입력 받기(총 5개의 공연을 상영 중인 것으로 작성함)
* 연령을 유아, 어린이, 학생, 성인, 노인으로 구분하여 연령에 따라 다른 가격 매기기
* 상영관 입력 받기(상영관 옵션 : 각 영화별로 1관, 2관, 3관 총 3개의 상영관이 존재하도록 함)
* 상영시간 입력 받기(상영시간 옵션 : 공연 별로 6개의 시간대, 상영관 별로 2개의 시간대를 둠)
* 좌석 예약 받기(좌석의 경우 실제 영화관처럼 열(가로, 행)과 번호(세로, 열)를 이용하여 입력을 받도록 작성하였음. 이 과정에서 2차원 배열을 활용)
* 결제금액 계산하기(연령에 따라 다른 가격을 부여했기 때문에 이를 토대로 각각의 티켓 유형별로 가격을 다르게 하여 총 금액 계산)
* 사용자가 예매한 결과를 표시하고 금액을 출력하기(예매 결과에는 선택한 공연 제목, 상영관, 상영시간, 구매한 티켓 수량, 예매한 좌석을 표시)
* 추가 옵션으로 사용자의 편의를 위해 이전 메뉴 재선택 의사를 묻는 메뉴를 출력하게 하고 이전 메뉴를 재선택 하거나 다음 메뉴로 넘어갈 수 있게 함
* 예매 완료 후 사용자가 완료된 예매를 취소할 수 있게 함

프로그램 작성 시 최대한 사용자가 보기 편하게 하기 위해 개행문자와 특수문자를 활용하였으며 통일성을 주기 위해 이전 메뉴 재선택 의사를 입력하는 키는 ‘r’로 통일하였다. 또한 메인 함수가 지나치게 길고 복잡해지는 것을 막기 위해 기능별로 함수를 구분하기 위해 노력했으며, 이에 따라 같은 변수를 서로 다른 함수에서 사용하는 경우가 발생하여 대부분의 변수가 전역 변수로 선언되었다.

본 프로그램에서 사용된 공연 제목은 필자가 임의로 지은 제목이며 실제 공연과 아무런 연관이 없음을 밝힌다.

1. **코드 및 주석**

// 극장 예매 프로그램

#include <stdio.h> // 표준 입출력 헤더파일 포함

#define COLS 10 // 좌석의 번호를 저장할 배열의 크기

#define ROWS 5 // 좌석의 열을 저장할 배열의 크기

#define TICKET 5 // 티켓의 종류가 5개(연령별로 구분)이므로 배열을 이용해 각각의 개수를 저장하기 위함

/\* 기호상수로 배열의 크기를 지정하여 배열의 크기를 변경하기 쉽게 하였다.\*/

int show\_get\_menu(); // 관람할 공연의 제목을 선택받는 함수

void time\_table(int a); // 영화 상영관과 시간을 입력받는 함수

void re\_movie\_time(); // 사용자가 잘못된 시간을 입력했을 때 시간을 다시 입력 받기 위한 함수

void price\_table(); // 연령별 티켓 가격 표 출력하는 함수

void qquantity(); // 구매할 티켓의 수를 입력받는 함수

void show\_seat(); // 좌석배치도 출력 및 좌석 선택받는 함수

void bill(); // 결제금액 계산 및 최종 결과 출력하는 함수

/\* 일반적으로 메인함수가 가장 우선하므로 메인함수보다 뒤에 오는 함수를 메인 함수에서 쓰기 위해서는 미리 함수의 원형을 정의해야 한다.

함수의 원형을 정의할 때는 반환형, 함수이름, 매개변수를 포함하여야 한다.\*/

int title = 0; // 공연 제목을 저장할 변수

int movie\_p = 0; // 공연 상영관을 저장할 변수

int movie\_t = 0; // 공연 상영시간을 저장할 변수

int All\_ticket = 0; // 구매할 모든 티켓 개수의 합을 저장할 변수

int quantity[TICKET] = {0}; // 티켓 종류별로 각각의 구매 개수를 저장하기 위한 배열

int i,j; // 좌석예매 시에 사용되는 반복제어변수

int seats[ROWS][COLS] = {0}; // 좌석을 저장하기 위한 배열, 배열의 모든 요소를 0으로 초기화함

/\* 좌석을 더 많이 두기 위해 2차원 배열을 이용하였다.

2차원 배열의 경우 요소들이 2차원적으로 배열되어 있으며 하나의 요소를 참조하려면 2개의 인덱스가 필요하다.

본 프로그램에서는 행의 크기가 5, 열의 크기가 10이므로 각 행마다 10개의 요소가 있다고 할 수 있다.

위의 변수 및 배열은 전역변수로 선언하여 서로 다른 함수 내에서 사용할 수 있게 하였다.

전역변수는 함수 외부에서 선언되는 변수로 범위는 소스 파일 전체에 해당한다.\*/

int main() // 반환형이 정수형이고 매개변수가 없는 메인함수의 선언부

{

char back = 0; // 공연 메뉴 재선택 의사를 저장할 변수

// 프로그램 안내

printf("☆★☆★☆★극장 예매 프로그램☆★☆★☆★\n\n");

// 반복문 이용하여 프로그램 수행

while(1){ //조건을 1로 하여 무한 반복되게 함, break; 만나면 탈출

title = show\_get\_menu(); // 공연메뉴 안내 및 공연메뉴 입력받는 함수를 호출하고 반환값 저장

/\* 함수 호출 시에는 반환형을 쓰지 않고 이름과 인수를 써서 호출한다.

반환값이 있는 경우 반환값을 저장하기 위해 변수 = 함수();의 형태로 사용한다.\*/

if(title == 6) // 만약 사용자가 공연메뉴 6(종료)을 선택한다면

{

break; // 반복문을 빠져나가 프로그램을 종료하는 메시지를 출력하고 종료

}

printf("\n☆귀하가 선택하신 공연은 "); // 사용자로부터 입력받은 공연 안내

switch(title) // switch문을 이용해 입력받은 번호에 따라 다르게 출력하기

{

case 1 : // 사용자가 공연 메뉴 1을 선택했을 때

printf("<빌 딩 숲>"); // 빌딩숲 출력

break;

case 2 : // 사용자가 공연 메뉴 2를 선택했을 때

printf("<아 스 팔 트>"); // 아스팔트 출력

break;

case 3 : // 사용자가 공연 메뉴 3을 선택했을 때

printf("<스 모 그>"); // 스모그 출력

break;

case 4 : // 사용자가 공연 메뉴 4를 선택했을 때

printf("<자 연 으 로>"); // 자연으로 출력

break;

case 5 : // 사용자가 공연 메뉴 5를 선택했을 때

printf("<나는자연인이다>"); // 나는자연인이다 출력

break; // switch문은 break;를 만나지 않으면 모든 케이스를 수행하므로 케이스마다 break;문 쓰기

}

printf(" 입니다.☆\n\n"); // 공연제목 출력

// 공연 재선택 의향 묻기

printf("공연 재선택 : r 입력\n"); // 공연 재선택을 안내

printf("상영관 선택 : r 이외의 키 입력\n"); // 상영관선택으로 넘어가는 법 안내

getchar(); // 문자형 변수를 입력받을 때 scanf();함수로 입력받지 않는 경우가 발생, 버퍼비우기

scanf("%c", &back); // 공연 재선택 의사 입력받기

printf("\n"); // 줄바꿈

if(back == 'r') // 사용자가 r을 입력할 경우

{

continue; // while문의 처음으로 돌아가 공연을 다시 선택 받기

// continue문은 현재 실행하고 있는 반복 과정의 나머지를 생략하고 다음 반복을 시작하게 만듦

}

time\_table(title); // 선택한 공연에 따라 상영관과 상영시간을 안내 및 입력받는 함수 호출

/\* 정수형 매개변수를 가진 함수이므로 정수형 인수를 줘야 하며 선택한 공연에 따라 상영관과 상영시간이 다르므로 인수로 사용자가 선택한 공연메뉴를 받는다.\*/

price\_table(); // 티켓 수량을 입력받기 전에 연령별 공연 요금을 안내하는 함수 호출

qquantity(); // 연령별로 티켓 수량을 입력받는 함수 호출

printf("좌석을 선택해주세요.\n\n"); // 좌석 선택 안내

show\_seat(); // 좌석 예약 함수 호출

bill(); // 총 금액 계산 및 예매 결과 출력하는 함수 호출

printf("\n☆예매가 완료되었습니다.☆\n\n"); // 예매 완료 안내

// 예매 취소 의사 묻기

printf("완료된 예 매 취 소 : x 입력\n"); // 예매 취소 안내하기

printf("예매 프로그램 종료 : x 이외의 키 입력\n"); // 예매 종료 안내하기

getchar(); // 문자형 변수 입력받기 위해 버퍼 비우기

scanf("%c", &cancle); // 예매취소의사 입력받기

if(cancle == 'x') // 사용자가 예매취소를 원할 때

{

for(i = 0; i < ROWS; i++) // 5번 반복

{

for(j = 0; j < COLS; j++) // 10번 반복

{

seats[i][j]; // 인덱스 값 주기

if(seats[i][j] == 1) // 좌석값이 1이면

seats[i][j] = 0; // 예약된 좌석 취소하기

}

}

printf("\n☆예매가 취소되었습니다.☆\n"); // 예매취소알림

}

break; // 반복문 빠져나가 예매 프로그램 종료하기

}

printf("\n☆예매 프로그램을 마칩니다.☆"); // 예매프로그램 종료 안내하기

return 0; // 메인함수에 0을 반환하여 끝내기

}

int show\_get\_menu() // 공연 메뉴를 표시하고 사용자로부터 메뉴를 선택받는 함수

// 반환값이 정수이고 매개변수는 없음

{

int n; // 공연 메뉴를 저장할 정수형 변수

printf("==============상영 중인 공연==============\n"); // 상영중인 공연 안내하기

printf("1. <빌 딩 숲>\n"); // printf()함수 이용하여 공연 제목 출력하기

printf("2. <아 스 팔 트>\n");

printf("3. <스 모 그>\n");

printf("4. <자 연 으 로>\n");

printf("5. <나는자연인이다>\n");

printf("6. <예 매 취 소>\n");

printf("==========================================\n");

// 공연메뉴 닫기

printf("★관람하실 공연을 선택해 주세요.\n"); // 공연 선택 안내하기

printf("관람하실 공연 : ");

scanf("%d", &n); // 사용자로부터 공연 메뉴 입력받아 n에 저장

while(n<=0 || n>6) // 사용자가 메뉴 이외의 수를 입력했을 때

{

printf("\n※메뉴에 없는 번호 입니다.※\n다시 입력해주세요.\n"); // 오류 메시지 출력

scanf("%d", &n); // 사용자로부터 다시 입력받기

}

return n; // 사용자로부터 입력받은 공연메뉴 반환하기

}

void time\_table(int a) // 공연 시간표를 출력하고 사용자로부터 상영관, 시간 입력받는 함수

{ // 반환값은 없으며 매개변수(공연 제목이 될 예정)는 정수형 변수

char back\_, back\_\_;

// 앞에서부터 차례대로 상영관, 상영시간 재입력 의사를 저장할 변수

// 문자형 변수이므로 변환명세 주의하기

printf("%d번 공연의 상영안내입니다.\n\n", a); // 공연 상영안내 출력하기

printf("===========================================\n");

printf(" 1관 2관 3관\n"); // 상영관 출력하기

switch(a){ // switch문 이용해 매개변수 값에 따라 다른 코드 수행하기

case 1 : // 매개변수의 값이 1일 때

printf("09:00-10:40 07:00-08:40 17:00-18:40\n");

// 1관 9시 2관 7시 3관 17시

printf("21:00-22:40 13:00-14:40 01:00-02:40\n");

// 1관 21시 2관 13시 3관 1시

break;

case 2 : // 매개변수의 값이 2일 때

printf("07:00-08:40 17:00-18:40 11:00-12:40\n");

// 1관 7시 2관 17시 3관 11시

printf("15:00-16:40 23:00-00:40 21:00-22:40\n");

// 1관 15시 2관 23시 3관 21시

break;

case 3 : // 매개변수의 값이 3일 때

printf("17:00-18:40 09:00-10:40 13:00-14:40\n");

// 1관 17시 2관 9시 3관 13시

printf("23:00-00:40 01:00-02:40 19:00-20:40\n");

// 1관 23시 2관 1시 3관 19시

break;

case 4 : // 매개변수의 값이 4일 때

printf("11:00-12:40 15:00-16:40 09:00-10:40\n");

// 1관 11시 2관 15시 3관 9시

printf("19:00-20:40 21:00-22:40 23:00-00:40\n");

// 1관 19시 2관 21시 3관 23시

break;

case 5 : // 매개변수의 값이 5일 때

printf("13:00-14:40 11:00-12:40 07:00-08:40\n");

// 1관 13시 2관 11시 3관 7시

printf("01:00-02:40 19:00-20:40 15:00-16:40\n");

// 1관 1시 2관 19시 3관 15시

break; //케이스마다 break문 반드시 입력

} // 상영관과 상영시간 자리를 맞추어 출력하여 상영 스케줄표를 표시함

printf("===========================================\n\n");

// 상영스케줄표 닫기

printf("★관람하실 상영관을 입력해주세요.\n"); // 상영 입력 안내하기

printf("상영관 : ");

scanf("%d", &movie\_p); // 상영관 입력받아서 정수형 변수 movie\_p에 저장하기

while(movie\_p < 1 || movie\_p > 3) // 제시된 상영관 외의 수를 입력했을 때 반복하기

{

printf("\n※선택 가능한 상영관이 아닙니다.※\n"); // 오류 메시지 출력

printf("다시 입력해주세요 : "); // 재입력 안내

scanf("%d", &movie\_p); // 상영관 재입력 받기

}

printf("\n☆귀하가 선택하신 상영관은 %d관 입니다.☆\n\n", movie\_p); // 선택한 상영관 안내하기

// 상영관 재선택 의사 묻기

printf("상 영 관 재선택 : r 입력\n"); // 상영관 재선택 안내

printf("상영시간 선택 : r 이외의 키 입력\n"); // 상영시간 선택 안내

getchar(); // back\_을 입력받지 않는 경우가 발생, 버퍼를 비움

// back\_이 문자형 변수이기 때문에 공백을 문자로 간주해 넘어가는 것으로 보임

scanf("%c", &back\_); // 사용자에게 상영관 재선택 의사를 입력받음

printf("\n"); // 줄바꿈

if(back\_ == 'r') // 사용자가 재선택 의사를 밝혔을 때

{

time\_table(a); // 상영표 출력 및 상영관, 시간 입력받는 함수 호출하여 상영관 재입력받기

}

While(1) // while문 이용하여 조건에 따라 상영시간 반복 입력받기

{

printf("★관람하실 시간을 입력해주세요.\n"); // 관람 시간 입력 안내

printf("(9시를 원하시면 9 의 형태로 시간을 입력해주세요.)\n");

// 입력예시 안내

printf("시간 : ");

scanf("%d", &movie\_t); // 사용자로부터 관람시간 입력받아 movie\_t에 저장하기

printf("\n"); // 줄바꿈

// 매개변수별(공연메뉴), 상영관별로 상영시간이 다르기에 조건문으로 각각을 다르게 처리

if(a == 1) // 매개변수가 1이면서

if(movie\_p == 1) // 1 상영관이면서

{

if(movie\_t != 9 && movie\_t != 21)

// 상영시간이 9시도 21시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간을 다시 입력받는 함수 호출

if(movie\_t == 9 || movie\_t == 21)

// 상영시간이 9시 혹은 21시이면

break;

// 반복문 탈출하여 사용자로부터 받은 상영시간 출력

}

}

}

else if(movie\_p == 2) // 2 상영관이면서

{

if(movie\_t != 7 && movie\_t != 13)

// 상영시간이 7시도 13시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 7 || movie\_t == 13)

// 상영시간이 7시 혹은 13시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else // 3 상영관이면서

{

if(movie\_t != 17 && movie\_t != 1)

// 상영시간이 17시도 1시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 17 || movie\_t == 1)

// 상영시간이 17시 혹은 1시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

if(a == 2) // 매개변수 값이 2일 때

if(movie\_p == 1) // 1 상영관이면서

{

if(movie\_t != 7 && movie\_t != 15)

// 상영시간이 7시도 15시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 7 || movie\_t == 15)

// 상영시간이 7시나 15시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else if(movie\_p == 2) // 2 상영관이면서

{

if(movie\_t != 17 && movie\_t != 23)

// 상영시간이 17시도 23시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 17 || movie\_t == 23)

// 상영시간이 17시나 23시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else // 3 상영관이면서

{

if(movie\_t != 11 && movie\_t != 21)

// 상영시간이 11시도 21시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 11 || movie\_t == 21)

// 상영시간이 11시나 21시면

{

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

}

if(a == 3) // 매개변수 값이 3일 때

if(movie\_p == 1) // 1 상영관이면서

{

if(movie\_t != 17 && movie\_t != 23)

// 상영시간이 17시도 23시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 17 || movie\_t == 23)

// 상영시간이 17시나 23시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else if(movie\_p == 2) // 2 상영관이면서

{

if(movie\_t != 9 && movie\_t != 1)

// 상영시간이 9시도 1시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 9 || movie\_t == 1)

// 상영시간이 9시나 1시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else // 3 상영관이면서

{

if(movie\_t != 13 && movie\_t != 19)

// 상영시간이 13시도 19시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 13 || movie\_t == 19)

// 상영시간이 13시나 19시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

if(a == 4) // 매개변수 값이 4일 때

if(movie\_p == 1) // 1 상영관이면서

{

if(movie\_t != 11 && movie\_t != 19)

// 상영시간이 11시도 19시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 11 || movie\_t == 19)

// 상영시간이 11시나 19시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else if(movie\_p == 2) // 2 상영관이면서

{

if(movie\_t != 15 && movie\_t != 21)

// 상영시간이 15시도 21시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 15 || movie\_t == 21)

// 상영시간이 15시나 21시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else // 3 상영관이면서

{

if(movie\_t != 9 && movie\_t != 23)

// 상영시간이 9시도 23시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 9 || movie\_t == 23)

// 상영시간이 9시나 23시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

if(a == 5) // 매개변수 값이 5일 때

if(movie\_p == 1) // 1 상영관이면서

{

if(movie\_t != 13 && movie\_t != 1)

// 상영시간이 13시도 1시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 13 || movie\_t == 1)

// 상영시간이 13시나 1시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else if(movie\_p == 2) // 2 상영관이면서

{

if(movie\_t != 11 && movie\_t != 19)

// 상영시간이 11시도 19시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 11 || movie\_t == 19)

// 상영시간이 11시나 19시면

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

else // 3 상영관이면서

{

if(movie\_t != 7 && movie\_t != 15)

// 상영시간이 7시도 15시도 아니면

{

while(1)

// 무조건 반복, break문 만나면 빠져나옴

{

re\_movie\_time();

// 상영시간 다시 입력받는 함수 호출

if(movie\_t == 7 || movie\_t == 15)

// 상영시간이 7시나 15시 일 때

break;

// 반복문 탈출, 입력받은 상영시간 출력

}

}

}

printf("☆귀하가 선택하신 상영시간은 %d시 입니다.☆\n\n", movie\_t);

// 선택 상영시간 안내

// 상영시간 재선택 의사 묻기

printf("상영시간 재선택 : r 입력 \n"); // 상영시간 재선택 안내하기

printf("티켓 수량 선 택 : r 이외의 키 입력\n"); // 티켓 수량 선택 안내하기

getchar(); // back\_\_입력받기 위해 버퍼 비우기

scanf("%c", &back\_\_); // 사용자로부터 상영시간 재선택 의사 입력받기

printf("\n"); // 줄바꿈

if(back\_\_ == 'r') // 사용자가 시간 재선택을 원할 때

{

continue; // 반복문 처음으로 돌아가 반복 시작

}

else // 사용자가 시간 재선택을 원하지 않을 때

break; // 반복문 빠져나가 티켓 수량 선택

}

}

void re\_movie\_time() // 사용자가 상영시간을 잘못 입력했을 때 재입력 받는 함수

{

printf("※선택 가능한 시간이 아닙니다.※\n"); // 오류 메시지 출력

printf("가능한 시간 중 하나를 선택해주세요. : "); // 재입력 안내

scanf("%d", &movie\_t); // 재입력받기

printf("\n"); // 줄바꿈

}

void price\_table() // 연령에 따른 공연 요금표 출력하는 함수

{

printf("공연 요금은 다음과 같습니다.\n\n"); // 공연 요금표 출력 안내

printf("===================요금표==================\n");

// 공연 요금표 출력

printf("1. 유아 7세 이하 4000원\n");

printf("2. 어린이 8세 이상 13세 이하 7000원\n");

printf("3. 학생 14세 이상 19세 이하 10000원\n");

printf("4. 성인 20세 이상 64세 이하 13000원\n");

printf("5. 경로우대 65세 이상 5000원\n");

printf("===========================================\n");

// 공연 요금표 닫기

}

void qquantity() // 사용자로부터 연령별 티켓 구매 수량 입력받는 함수

{

char backback; // 티켓 수량 재입력 의사 저장할 함수

printf("\n★티켓을 각각 몇 장 구매하시겠습니까?\n\n다음과 같이 입력해주세요\n");

printf("유아 어린이 학생 성인 경로우대 : 1 1 1 1 1\n"); // 티켓 수량 입력 방식 안내하기

do //반복문을 이용해 처리하기

{

printf("유아 어린이 학생 성인 경로우대 : "); // 티켓 유형 출력하기

scanf("%d %d %d %d %d", &quantity[0], &quantity[1], &quantity[2], &quantity[3], &quantity[4]);

// 유형별 티켓을 입력받고 배열에 저장하기

All\_ticket = quantity[0] + quantity[1] + quantity[2] + quantity[3] + quantity[4];

// 모든 티켓 수량 더하기, 티켓 수량이 0이거나 최대 수량보다 많으면 오류 메시지 출력하기 위함

while(All\_ticket <= 0 || All\_ticket >50) // 구매할 모든 수량이 0이거나 0보다 작다면

{

printf("\n※예매 가능한 좌석 수는 최소 1석 최대 50석입니다.※\n"); // 오류메시지 출력

printf("구매하실 티켓 수량을 다시 입력해주세요.\n");

// 재입력 안내

qquantity(); // 함수를 호출해 다시 입력받기

}

printf("\n☆귀하가 입력하신 티켓 수량은 다음과 같습니다.☆\n"); // 티켓 유형별 수량 안내

printf("유 아 : %2d\n", quantity[0]); // 유아 티켓 수량 안내

printf("어 린 이 : %2d\n", quantity[1]); // 어린이 티켓 수량 안내

printf("학 생 : %2d\n", quantity[2]); // 학생 티켓 수량 안내

printf("성 인 : %2d\n", quantity[3]); // 성인 티켓 수량 안내

printf("경로우대 : %2d\n\n", quantity[4]); // 경로우대 티켓 수량 안내

// 티켓 수량 재입력 의사 묻기

printf("티켓 수량 재입력 : r 입력\n"); // 티켓 수량 재입력 안내

printf("좌 석 선 택 : r 이외의 키 입력\n"); // 좌석선택 안내

getchar(); // 문자형 변수 입력받기 위해 버퍼비우기

scanf("%c", &backback); // 티켓 수량 재입력 의사 입력받기

printf("\n"); // 줄바꿈

}while(backback == 'r'); // 사용자가 티켓 수량 재입력 의사를 보이는 동안 반복하기

}

void show\_seat() // 좌석 배치도를 표시하고 좌석 선택받는 함수

{

// 변수 선언

int row = 0; // 좌석 열 입력받아 저장할 변수

int ans2 = 0; // 좌석 번호를 입력받아 저장할 변수

int k = 0; // 좌석 선택을 티켓 수량만큼 반복하기 위한 변수, 0으로 초기화

char backback\_ = 0; // 좌석 재입력 의사 받는 변수

do // do while문을 이용해 반복

{

printf("=================좌석배치도===============\n"); // 좌석 배치도 출력

printf(" 1 2 3 4 5 6 7 8 9 10\n"); // 좌석 번호 표시

for(i = 0; i < ROWS; i++) // 5번 반복(5행)

{

printf("%d", i+1); // 좌석 열 표시

for(j = 0; j < COLS; j++) // 10번 반복(10열)

{

printf(" %d ", seats[i][j]); // 좌석 출력

}

printf("\n"); // 행마다 줄바꿈

} // 5행 10열 총 50번 반복

printf("==========================================\n"); //좌석 배치도 출력 완료 알림

printf("\n"); // 줄바꿈

printf("★몇번째 열(1~5 중 택)을 선택하시겠습니까? "); // 좌석의 열 선택 안내

scanf("%d", &row); // 좌석 열을 입력받기(행열 중 행에 해당)

printf("\n"); // 줄바꿈

while(row <= 0 || row > ROWS) // 입력받은 열이 0이거나, 0보다 작거나, 5보다 크면 재선택

{

printf("※열은 1열부터 5열까지 있습니다.※\n"); // 입력 가능한 열의 수 안내

printf("희망하시는 열을 다시 입력해 주세요. "); // 재입력 안내

scanf("%d", &row); // 열 재입력 받기

printf("\n"); // 줄바꿈

}

printf("★몇번째 좌석(1~10 중 택)을 선택하시겠습니까? "); // 좌석 번호 선택 안내

scanf("%d", &ans2); // 좌석 번호 입력 받기(행열 중 열에 해당)

printf("\n"); // 줄바꿈

while(ans2 <= 0 || ans2 > COLS) // 좌석 본호가 0이거나 0보다 작거나 10보다 크면 반복

{

printf("※좌석은 1부터 10까지만 선택 가능합니다.※\n"); // 입력 가능한 좌석 수 안내

printf("희망하시는 좌석 번호를 다시 입력해 주세요. "); // 재입력 안내

scanf("%d", &ans2); // 좌석 번호 재입력 받기

printf("\n"); // 줄바꿈

}

if(ans2 > 0 || ans2 < 11) // 좌석 번호가 1부터 10사이이고

if(row > 0 || row < 6) // 좌석 열이 1부터 5 사이일 때

{

if(seats[row-1][ans2-1] == 0) // 좌석 값이 0이면

{

printf("☆귀하가 선택하신 좌석은 %d열 %d번입니다.☆\n\n", row, ans2);

// 좌석 예약 완료 메시지 출력

// 좌석 재선택 의사 묻기

printf("좌 석 재 선 택 : r 입력\n");

// 좌석 재선택 안내

printf("다음으로 넘어가기 : r 이외의 키 입력\n"); // 다음 단계 진행안내

getchar(); // 문자형 변수 입력받기 위해 버퍼비우기

scanf("%c", &backback\_); // 사용자로부터 좌석 재선택 의사 입력받기

printf("\n"); // 줄바꿈

if(backback\_ == 'r') // 사용자가 r을 입력했을 때

{

continue; // 다음 반복 시작하기

}

else // 사용자가 r 이외의 키를 눌렀을 때

seats[row-1][ans2-1] = 1;

// 사용자가 좌석을 선택했으므로 배열 값에 1 넣기

}

else // 좌석 값이 0이 아니면

{

printf("※이미 예약된 자리입니다.※\n\n"); // 예약 불가능 안내

printf("좌석을 다시 선택해주세요.\n"); // 좌석 다시 선택받기

continue; // k++를 생략하고 다음 반복을 시작하게 함

}

}

k++; // 한번 완료할 때마다 k 1씩 증가

}while(k != All\_ticket);

// 티켓 총수량과 k가 일치하면 반복문을 빠져나옴(티켓수량만큼 반복)

}

void bill() // 결제 금액 계산 및 예매결과, 결제 금액 출력하는 함수

{

int price; // 결제 금액 계산하기 위한 정수형 변수

printf("==================예매결과================\n\n"); // 예매내역 출력하기

printf(" ★선택하신 공 연★\n\n"); // 선택한 공연 안내하기

switch(title) // switch문 이용해 title변수에 입력된 수에 따라 제목 출력하기

{

case 1 : // 공연제목메뉴 1을 입력했을 때

printf(" <빌 딩 숲>\n"); // 빌딩숲 출력하기

break;

case 2 : // 공연제목메뉴 2를 입력했을 때

printf(" <아 스 팔 트>\n"); // 아스팔트 출력하기

break;

case 3 : // 공연제목메뉴 3을 입력했을 때

printf(" <스 모 그>\n"); // 스모그 출력하기

break;

case 4 : // 공연제목메뉴 4를 입력했을 때

printf(" <자 연 으 로>\n"); // 자연으로 출력하기

break;

case 5 : // 공연제목메뉴 5를 입력했을 때

printf(" <나는자연인이다>\n"); // 나는자연인이다 출력하기

break; // switch문에서는 break;문을 쓰지 않으면 break;를 만날 때까지 계속 진행함

}

printf("\n ★선택하신 상 영 관★\n\n %3d 관\n", movie\_p); // 선택한 상영관 안내하기

printf("\n ★선택하신 상영시간★\n\n %3d 시\n", movie\_t); // 선택한 시간 안내하기

printf("\n ★선택하신 좌 석★\n\n"); // 선택한 좌석 안내하기

// 좌석 값이 1인 좌석 출력하기 위한 반복문

for(i = 0; i < ROWS; i++) // 5번 반복

{

for(j = 0; j < COLS; j++) // 10번 반복

{

seats[i][j]; // 인덱스 값 주기

if(seats[i][j] == 1) // 좌석값이 1이면

printf(" %d열%2d번\n", i+1, j+1); // 예약한 좌석 출력하기

}

}

printf("\n ★선택하신 티켓 매수★\n\n"); // printf()함수로 티켓 매수 안내하기

printf("유아 어린이 학생 성인 경로우대\n"); // 티켓 유형 출력

printf("%d %d %d %d %d\n", quantity[0], quantity[1], quantity[2], quantity[3], quantity[4]);

// 선택한 티켓 수량 출력

printf("\n===========================================\n\n"); // 예매내역 닫는 줄 출력

price = quantity[0]\*4000 + quantity[1]\*7000 + quantity[2]\*10000 + quantity[3]\*13000 + quantity[4]\*5000;

// 유형별로 가격을 다르게 하여 배열(유형별 티켓 수)과 가격을 곱한 것을 더해 티켓 총 가격 계산하기

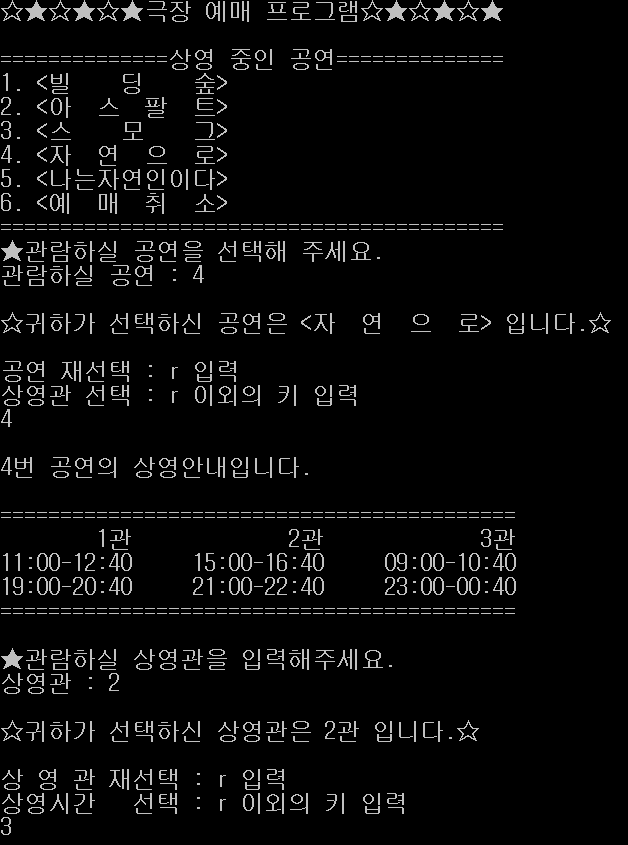
price\_table(); // 요금표 함수 호출하여 요금표 표시하기

printf("\n총금액 : %d\n", price); // 총 금액 출력

}

1. **정상 실행화면**

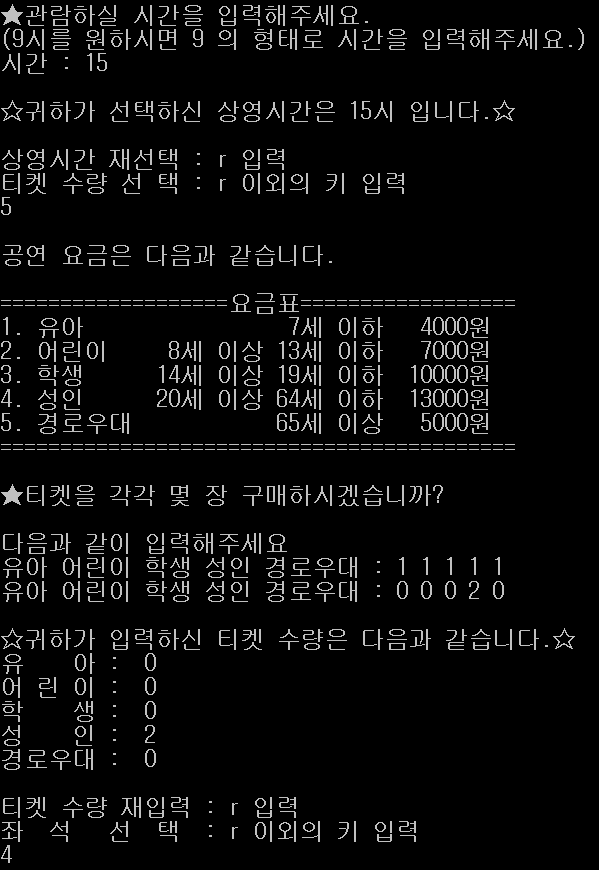


재입력하지 않았을 때의 실행화면

프로그램 이름을 출력하고 이어서 공연 메뉴를 출력한다.

공연메뉴를 입력받은 후에는 재입력 의사를 묻고, 여기서는 재입력 의사를 받지 않았기에 상영시간표를 출력, 상영관을 입력받는다.

상영관을 입력받은 후 재입력 의사를 묻는데 여기서 사용자는 재입력 의사가 없으므로 r 이외의 키를 입력하였다.

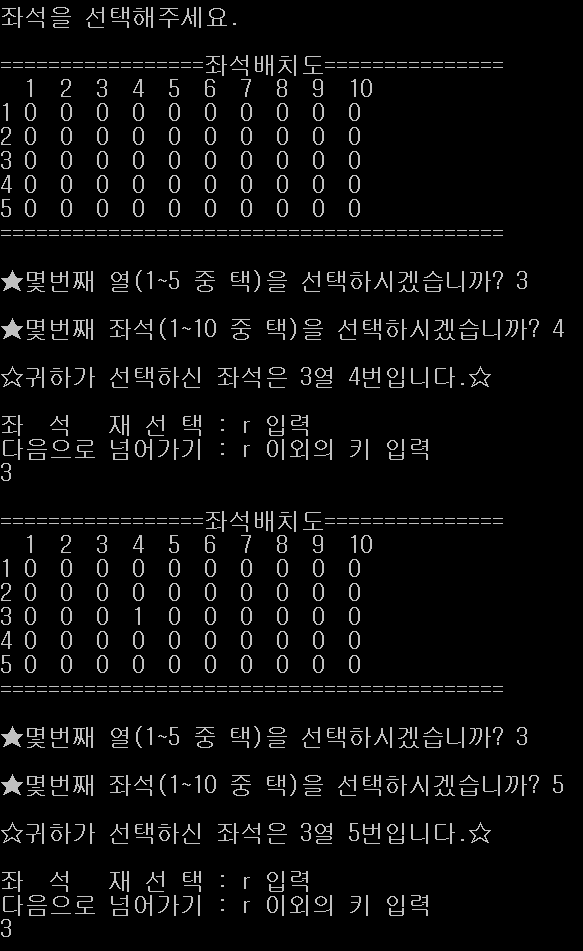
****

상영시간을 입력받은 마찬가지로 재입력 의사를 묻고 재입력의사가 없으면 공연 요금을 안내한다.

연령에 따른 공연 요금표를 먼저 제시한 다음 티켓 수량을 입력받는다.

티켓 수량은 각 유형별로 다르게 입력받는다.

입력받은 티켓 수량을 출력하여 확인을 받고 티켓 수량 재입력 의사를 묻는다.

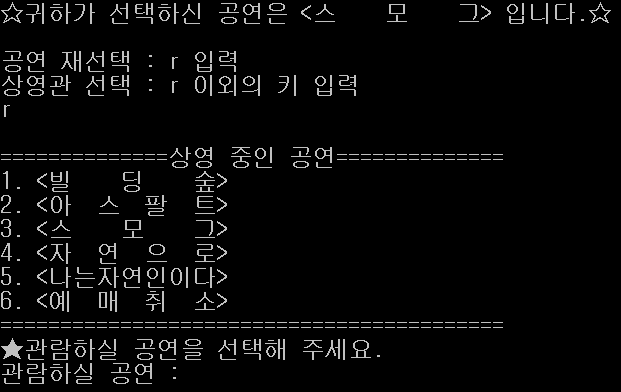
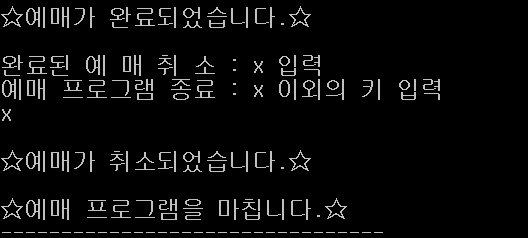
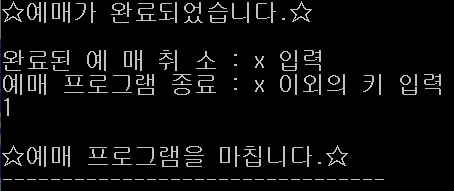
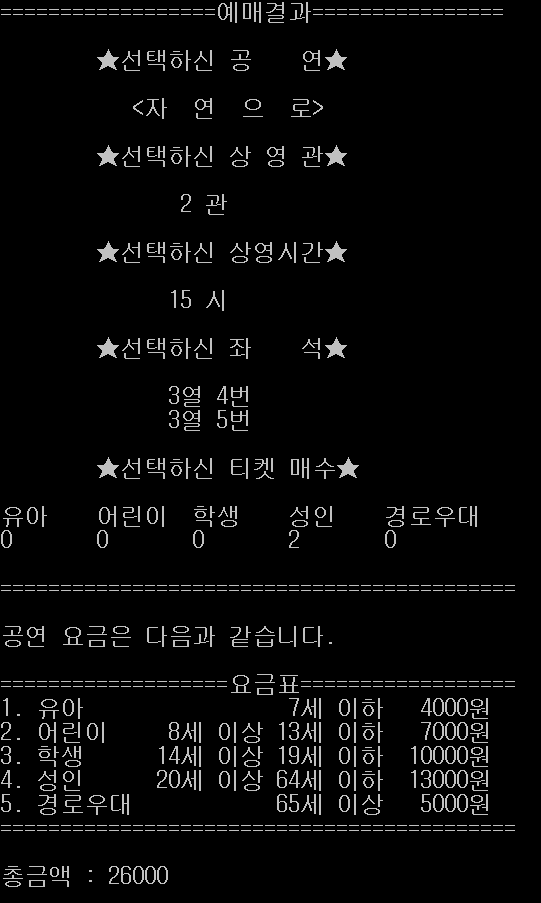


좌석을 예약 시에는 좌석 배치도를 먼저 출력하고 좌석 열, 번호 순으로 입력 받는다.

사용자가 입력한 좌석 수에 따라 각각의 좌석의 열과 번호를 입력받는다.

여기서 사용자는 티켓을 2매 구매하기로 했기 때문에 좌석 선택은 2번 반복된다.

좌석선택이 끝날 때마다 재선택 의사를 묻고 재선택 의사가 없으면 다음으로 넘어간다.

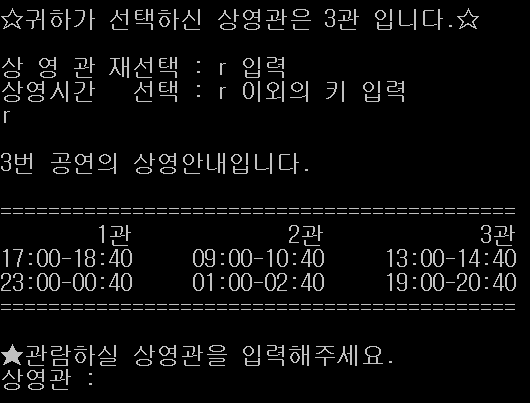
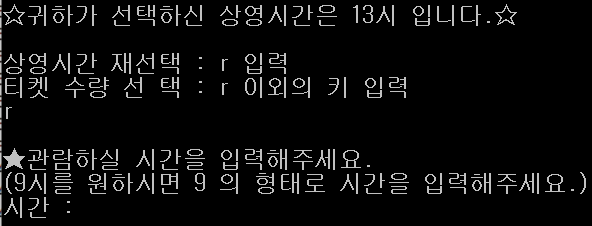
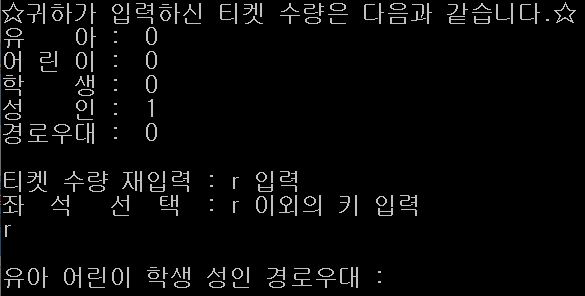
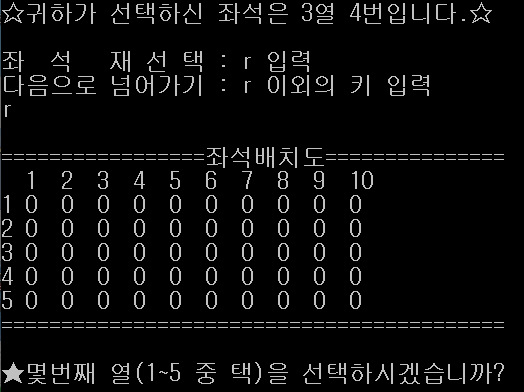


좌석 선택이 끝나면 예매 결과를 출력한다.

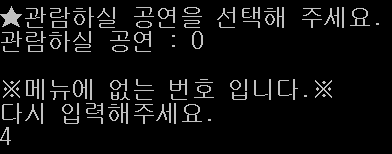
공연 제목, 상영관, 상영시간, 좌석, 선택한 티켓 매수를 출력하며 결제 금액 확인을 위해 공연 요금표를 다시 한 번 출력한다.

공연 요금표를 출력한 후에 결제 금액을 출력하고 예매 취소 의사를 물은 후 답을 받고 프로그램을 마친다

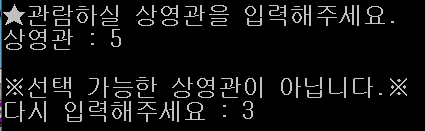
왼쪽의 실행화면은 사용자가 이전 메뉴 재입력을 택했을 때의 화면이다.

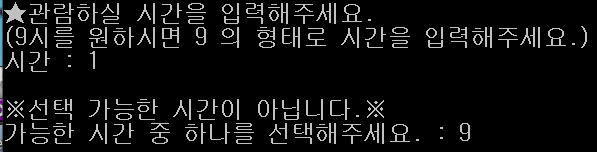


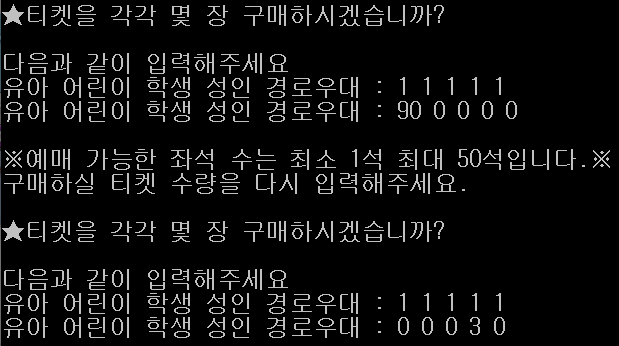
1. **오류 발생 시 실행화면**

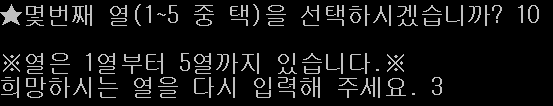


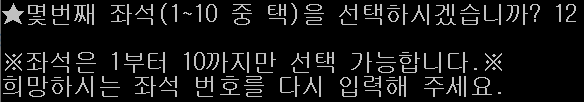
왼쪽의 실행화면은 오류 메시지와 관련된 실행화면이다.

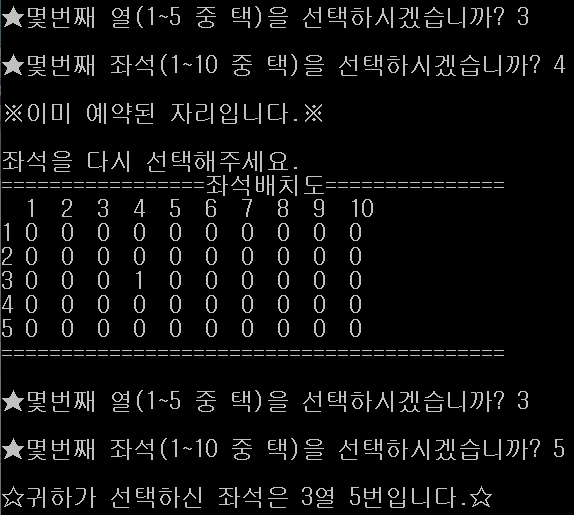












1. **보고서를 마치며**

지금까지 작성한 코드 중 가장 길고 다양한 기능을 담은 프로그램이기 때문에 완성하였을 때 뿌듯함도 있었으나, 아직 미숙하기 때문에 아쉬운 부분도 많은 것이 사실이다.

예를 들어, 공연별로 그리고 상영관별로 상영시간이 상이하기 때문에 상영시간 오류 처리 부분에서 코드가 굉장히 길고 복잡해졌는데 앞으로 프로그래밍을 더 배우고 지식을 쌓아 이러한 코드를 간결하게 할 수 있는 방법을 찾아 추후 코드를 수정할 예정이다. 마찬가지로, 반복되는 코드도 최대한 줄이는 방향으로 보완할 것이다.

또한 앞서 설명했던 것처럼 사용자가 최대한 보기 편하게 하기 위해 특수문자와 개행문자를 활용하였는데 더욱 편리한 프로그램을 만들기 위해서는 효과적인 출력 방안을 계속해서 고민해야 할 것 같다.

더불어 추가로 구현하고 싶었던 기능이 있는데 완벽히 구현할 수 없어 기능을 포기한 점도 있기에 느끼는 아쉬움이 크다.

스스로 프로그램이 갖추어야 할 기능을 정의하여 그것을 실현하는 과정에서 많은 시행착오가 있었고 부족한 점도 많지만 새로운 영역을 배우며 미력하나마 나만의 프로그램을 만든다는 점에서 느끼는 점이 많았던 것 같다.

앞으로 C언어를 배워나가면서 본 프로그램의 코드를 계속해서 수정하여 더욱 완성도 높은 프로그램을 만들 수 있도록 노력하고 싶다.

