

# 소스코드 및 주석

/\*

[과제#7] 나만의 티켓 예매 프로그램

문제: 이번주 실습#2로 수행하는 극장 예약 프로그램에 대해 이해하고,

단순히 배열을 이용한 좌석 예약 기능 뿐만 아니라

지금까지 배운 내용을 바탕으로 공연티켓 예매를 진행할 수 있는

기능을 각자 개별적으로 고민해서 정의하고, 이를 구현한다.

포함해야 할 기능

1. 공연제목 선택

2. 성인/청소년 몇 장 구매할 것인지

3. 장소 및 공연시간 선택

4. 좌석 예약

5. 결제금액 계산

6. 최종 선택한 내용 및 결제금액 결과 출력

알고리즘 :

3 x 4 x 100배열로 각 예매표를 표현한다.

-> 3중 for문으로 모든 표의 예매상태를 0으로 세팅한다.(초기 한 번만)

-> 선택가능한 제목, 시간을 표시해준다

-> 제목과 시간을 입력받아 그 시간의 상영관 예약 상태를 전부 출력한다.(잘못 입력 시 다시 선택)

-> 청소년, 성인 티켓 장 수 입력받기 (예매 가능 숫자보다 많이 입력했을 경우 다시 선택)

-> 결제하고 거스름돈 주기 (돈이 부족하면 pay는 0이다)

-> 예약하고 선택한 각각 자리를 1로 바꾸기 (위에서 결제가 되지 않았다면 예약 안됨)

\*/

#include <stdio.h>

#include <stdlib.h>

int ticket[3][4][100];

void clear\_meun() {

//3중 for문으로 모든 표의 예매상태를 0으로 세팅한다.(초기 한 번만)

for (int i = 0; i <3; i++) {

for (int j = 0; j <4; j++) {

for (int k = 0; k < 100; k++) {

ticket[i][j][k] = 0;

}

}

}

}

void print\_meun() {

//선택가능한 제목, 시간을 표시해준다

printf("1. 해리 포터 \*\* 1 :9시A, 2 :12시A, 3 :1시B, 4 :5시B\n");

printf("2. 볼드모트 \*\* 1 :9시B, 2 :12시B, 3 :1시A, 4 :5시A\n");

printf("3. 제임스 포터 \*\* 1 :9시C, 2 :12시C, 3 :1시C, 4 :5시C\n");

}

int show\_point(int title, int time) {

//제목과 시간을 입력받아 그 시간의 상영관 예약 상태를 전부 출력한다.(잘못 입력 시 다시 선택)

int avail\_seat = 0;

printf("이용 가능한 좌석은 숫자, 불가능한 좌석은 x입니다.\n");

for (int i = 0; i < 100; i++) {

if (ticket[title][time][i] == 0) {

printf("%3d \*\*\*", i+1);

avail\_seat++;

}

else if (ticket[title][time][i] == 1) {

printf(" x \*\*\*");

}

if ((i+1) % 10 == 0)

printf("\n");

}

return avail\_seat;

}

int pay\_ticket(int kid, int adult) {

//결제하고 거스름돈 주기(돈이 부족하면 pay는 0이다)

int used\_money = kid \* 5000 + adult \* 10000;

printf("전체 금액은 %d입니다.\n", used\_money);

int input\_money;

printf("지폐를 투입해주세요.");

scanf\_s("%d", &input\_money);

int total\_change = input\_money - used\_money;

if (total\_change < 0) {

printf("투입한 금액이 필요한 금액보다 적어 예약이 불가능합니다.\n");

return 0;

}

else {

printf("총 거스름돈 %d 원\n", total\_change);

int b\_50000 = total\_change / 50000;

total\_change = total\_change - b\_50000 \* 50000;

printf("50000원짜리 지폐 %d 개\n", b\_50000);

int b\_10000 = total\_change / 10000;

total\_change = total\_change - b\_10000 \* 10000;

printf("10000원짜리 지폐 %d 개\n", b\_10000);

int b\_5000 = total\_change / 5000;

total\_change = total\_change - b\_5000 \* 5000;

printf("5000원짜리 지폐 %d 개\n", b\_5000);

int b\_1000 = total\_change / 1000;

total\_change = total\_change - b\_1000 \* 1000;

printf("1000원짜리 지폐 %d 개\n", b\_1000);

return 1;

}

}

void reserve\_ticket(int title, int time, int kid, int adult, int pay) {

//예약하고 선택한 각각 자리를 1로 바꾸기(위에서 결제가 되지 않았다면 예약 안됨)

if (pay == 1) {

int num\_of\_ticket = 1;

while (num\_of\_ticket <= kid + adult) {

printf("%d개의 티켓 중 %d번째의 티켓의 위치를 입력해주세요", kid + adult, num\_of\_ticket);

int reserve\_point;

scanf\_s("%d", &reserve\_point);

if (ticket[title][time][reserve\_point-1] == 0) {

ticket[title][time][reserve\_point-1] = 1;

printf("%d번으로 예약되었습니다.\n", reserve\_point);

num\_of\_ticket++;

}

else {

printf("%d번은 이미 예약되었습니다.\n", reserve\_point);

}

}

}

}

int main() {

int title, time;

int kid, adult;

int pay = 0;

int end;

clear\_meun();

while (1) {

print\_meun();

printf("원하시는 영화 제목 번호를 입력해주세요");

scanf\_s("%d", &title);

printf("원하시는 영화 시간 번호를 입력해주세요");

scanf\_s("%d", &time);

if (title >= 1 && title <= 3 && time >= 1 && time <= 4) {

int avail\_num = show\_point(title, time);

printf("\n예약하실 청소년의 티켓 수를 입력해주세요");

scanf\_s("%d", &kid);

printf("\n예약하실 성인의 티켓 수를 입력해주세요");

scanf\_s("%d", &adult);

//청소년, 성인 티켓 장 수 입력받기 (예매 가능 숫자보다 많이 입력했을 경우 다시 선택)

if (avail\_num > kid + adult) {

pay = pay\_ticket(kid, adult);

reserve\_ticket(title, time, kid, adult, pay);

printf("종료 : 0 , 계속 : 0외의 숫자");

}

else {

printf("예약가능한 자리가 부족합니다. 다른 제목, 시간을 선택해주세요");

printf("종료 : 0 , 계속 : 0외의 숫자");

}

}

else {

printf("제목은 1-3이고 시간은 1-4입니다\n 다시 입력해주세요\n");

printf("종료 : 0 , 계속 : 0외의 숫자");

}

scanf\_s("%d", &end);

if (end == 0)

break;

}

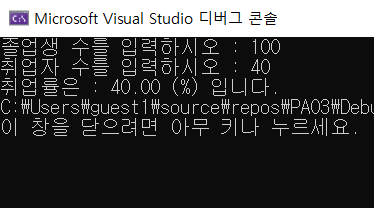
return 0;

}

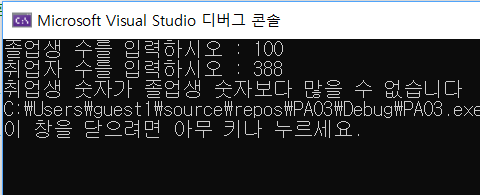
# 실행파일

실습 3

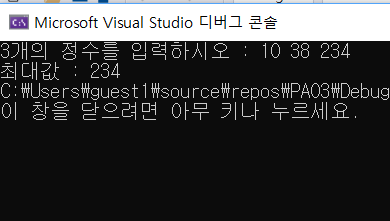
정상적으로 작동될 경우



졸업생보다 취업자 수가 많을 경우

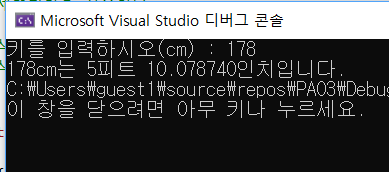


실습 4

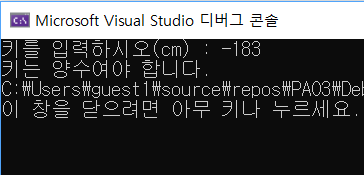


실습5

정상적으로 작동할 경우

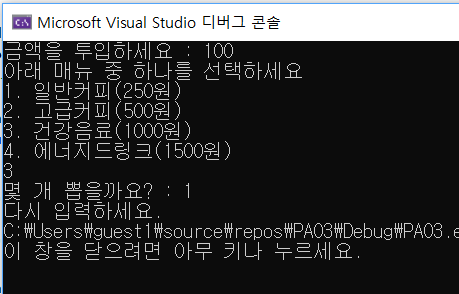


키에 0보다 작거나 같은 값을 넣을 경우

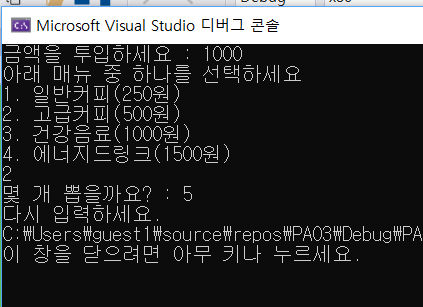


과제1

1. 금액이 적을 경우



1. 입력한 돈보다 사용금액이 클 경우



1. 정상적으로 돈을 넣어 작동될 경우

