ㄱ. 예약

// 스케쥴을 예약한다.

void AirlineBook::book() {

int s, ask;

string bookName;

int seatNo, plusHydrate, meal;

s = Console::getScheduleMenu(); // 사용자가 선택한 스케쥴을 입력 받는다.

view(s); // 스케쥴 s의 예약 상황을 출력한다.

seatNo = Console::getSeatNo(); // 좌석 번호를 입력받는다.

if (sche[s - 1].booling(seatNo) == true) { //입력한 좌석번호에 저장되어있는 정보가 없다면

bookName = Console::getName(); // 예약자의 이름을 입력받는다.

plusHydrate = Console::hydrate(); //위탁 수화물 갯수를 입력받는다

cout << "기내식을 추가하시겠습니까?" << endl; //기내식 추가여부 확인

cout << "\t" << "Yes : 1 No : 2 " << endl;

cin >> ask;

switch (ask) {

case 1: //추가하는 경우

meal = Console::food();

break;

case 2: //추가 하지 않는 경우

cout << "기내식을 추가하지 않습니다" << endl;

meal = 0;

}

sche[s - 1].book(seatNo, bookName, plusHydrate, meal);// 해당 스케쥴 예약

sche[s - 1].fee(plusHydrate, meal); //해당 스케쥴의 추가요금 계산

cout << "예약 완료되었습니다" << endl;

}

else { //해당 좌석에 이미 예약이 있는 경우

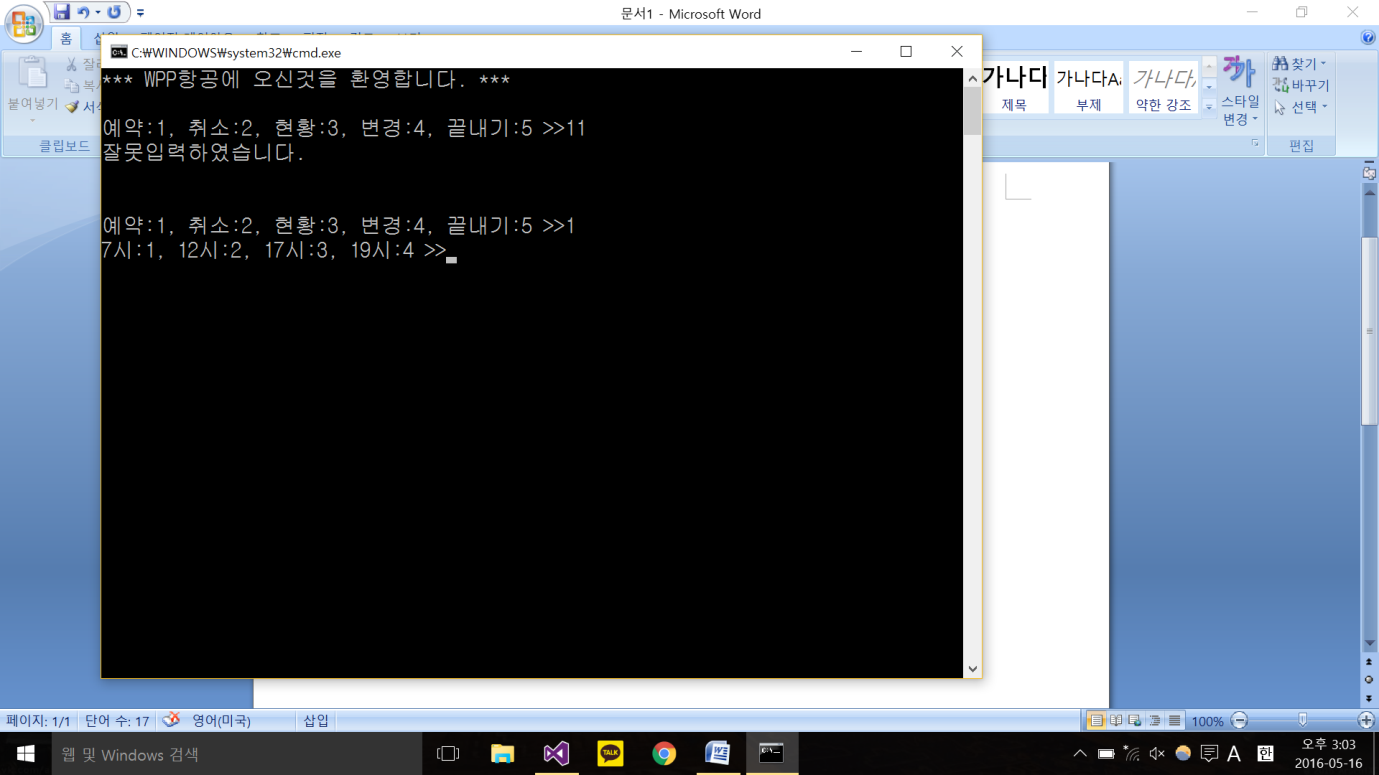
cout << "예약이 찬 좌석입니다" << endl;

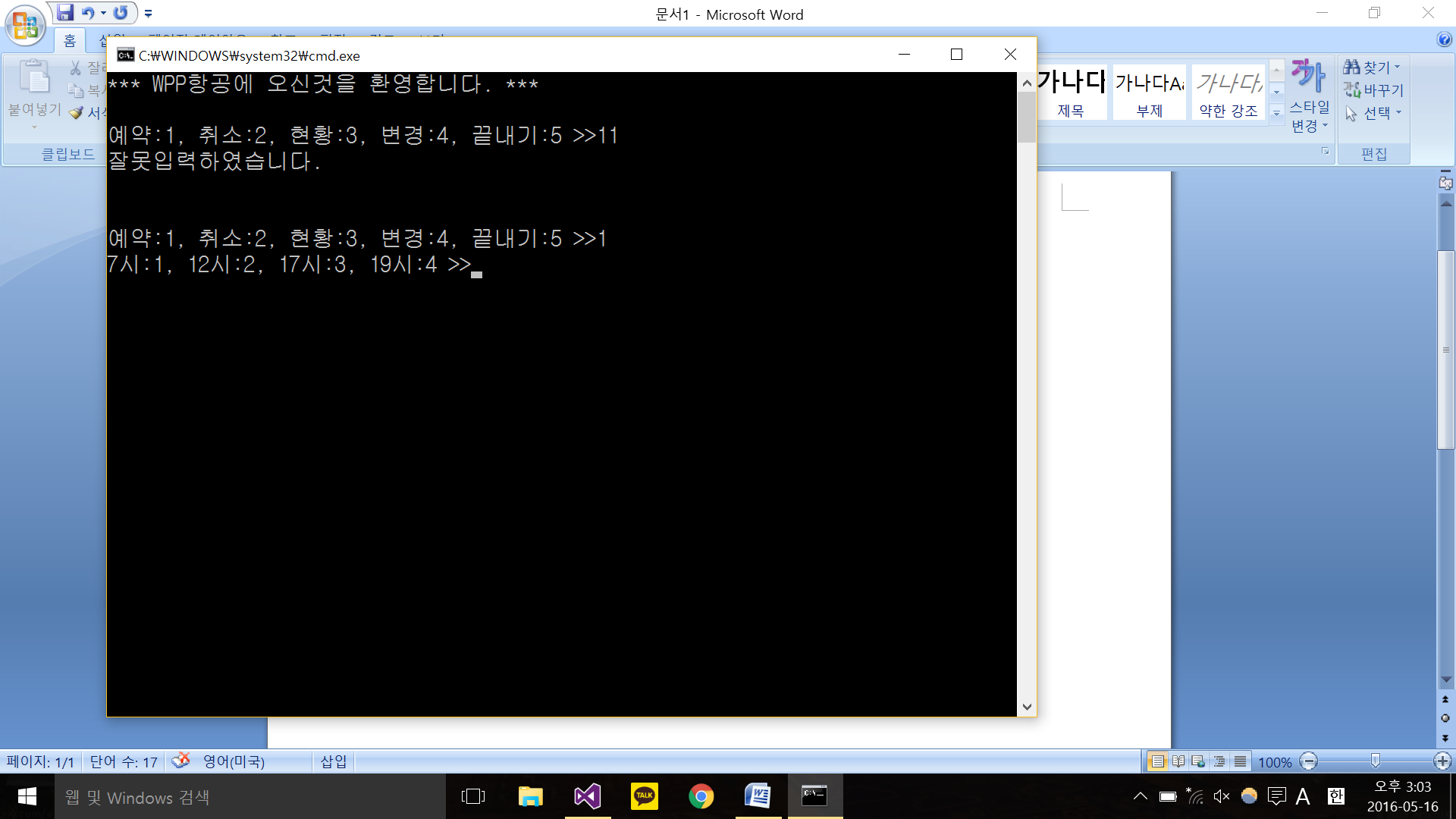
cout << "메인으로 돌아갑니다" << endl;

}

} A. 비행기 시간 추가

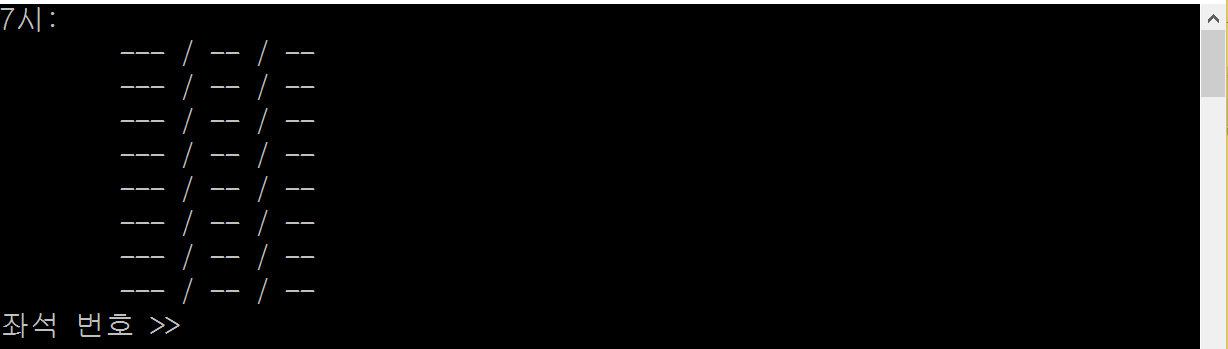
19시 추가





B. 좌석출력에 변화

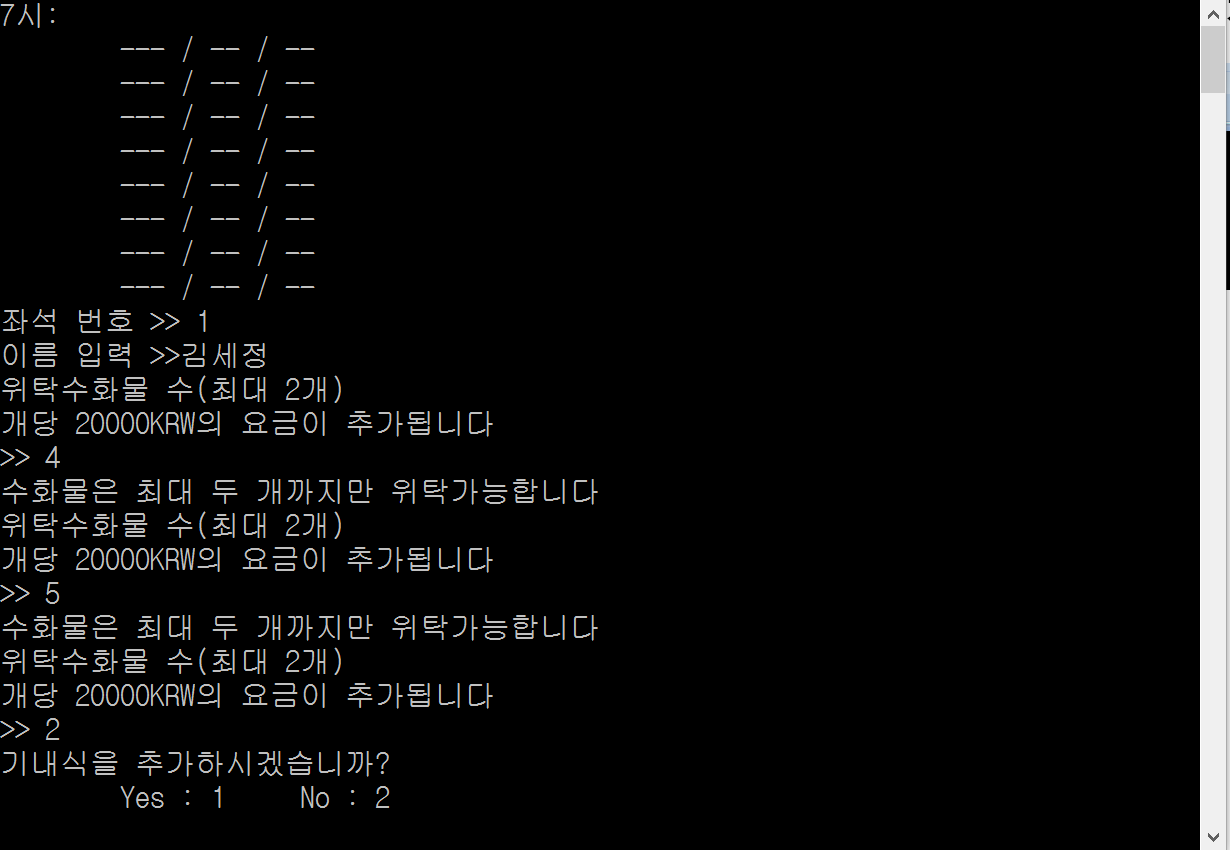
이름 / 수화물수 / 기내식선택메뉴 로 세로로 출력



C. 예약시 수화물 추가

수화물은 최대 두 개까지 추가 가능/ 개당 20000원의 추가요금

2개 이상의 수화물을 입력받을 경우 재입력을 요구.



int Console::hydrate() { //hydrate(): 위탁수하물 갯수 입력 함수

cout << "위탁수화물 수(최대 2개)" << endl;

cout << "개당 20000KRW의 요금이 추가됩니다" << endl;

cout << ">> ";

int num;

cin >> num;

if ((-1 < num) && (num < 3)) { //0~2 만의 입력을 받는다.

return num;

}

else { //그거 아니면 거부하고 재입력 요구.

cout << "수화물은 최대 두 개까지만 위탁가능합니다" << endl;

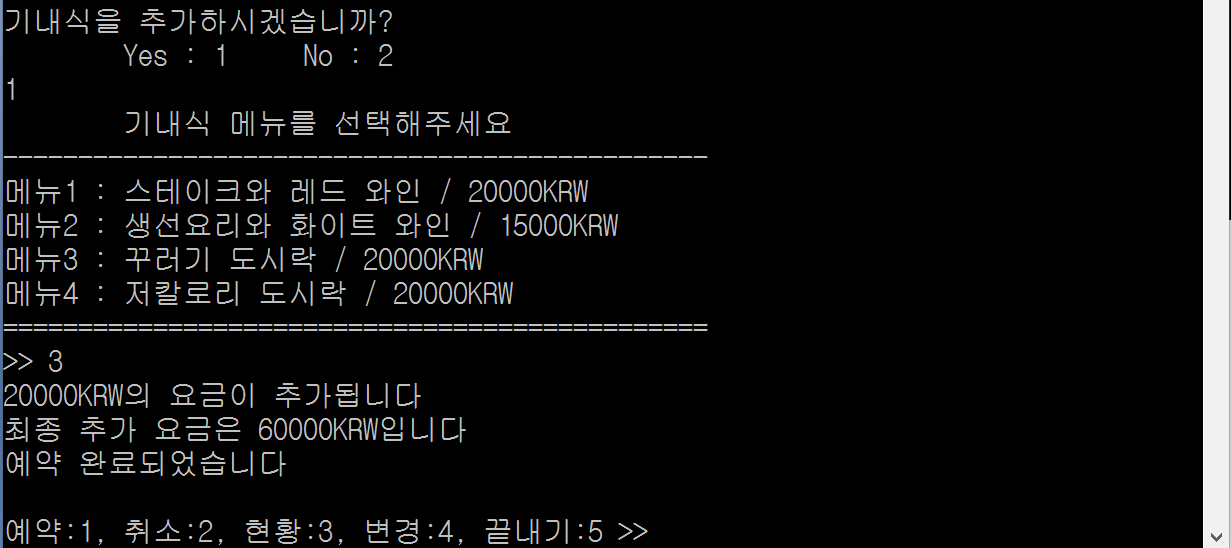
hydrate();

}

}

D. 기내식 추가

기내식 메뉴는 네 개로 그 중 하나의 메뉴를 선택한다



int Console::food() { //기내식 메뉴선택 함수

cout << "\t기내식 메뉴를 선택해주세요" << endl;

cout << "-----------------------------------------------" << endl;

cout << "메뉴1 : 스테이크와 레드 와인 / 20000KRW" << endl;

cout << "메뉴2 : 생선요리와 화이트 와인 / 15000KRW" << endl;

cout << "메뉴3 : 꾸러기 도시락 / 20000KRW" << endl;

cout << "메뉴4 : 저칼로리 도시락 / 20000KRW" << endl;

cout << "===============================================" << endl;

cout << ">> ";

int menu\_choice;

cin >> menu\_choice;

switch (menu\_choice) {

case 1: //메뉴1: 스테이크와 레드 와인

cout << "20000KRW의 요금이 추가됩니다 " << endl;

return menu\_choice;

break;

case 2: //메뉴2 : 생선요리와 화이트 와인

cout << "15000KRW의 요금이 추가됩니다" << endl;

return menu\_choice;

break;

case 3: //메뉴3 : 꾸러기 도시락

cout << "20000KRW의 요금이 추가됩니다" << endl;

return menu\_choice;

case 4: //메뉴4 : 저칼로리 도시락

cout << "20000KRW의 요금이 추가됩니다" << endl;

return menu\_choice;

break;

default: //그 외의것 입력하는 경우

cout << "잘못 입력하셨습니다" << endl;

}

}

E. 총 추가요금 표시

수화물 한 개 당 20000원, 선택한 기내식의 값의 합

void Console::fee(int hydrates, int food) { //추가 요금들을 계산 하는 함수

int menuprice[4] = { 20000, 15000, 20000, 20000 }; //기내식 메뉴 네가지의 가격

int fee;

if (food == 0) { //기내식을 신청하지 않았을경우

fee = hydrates \* 20000; //신청한 수하물 수x20000원으로 계산

}

else if (food < 5 && food>0) { //기내식을 신청한 경우

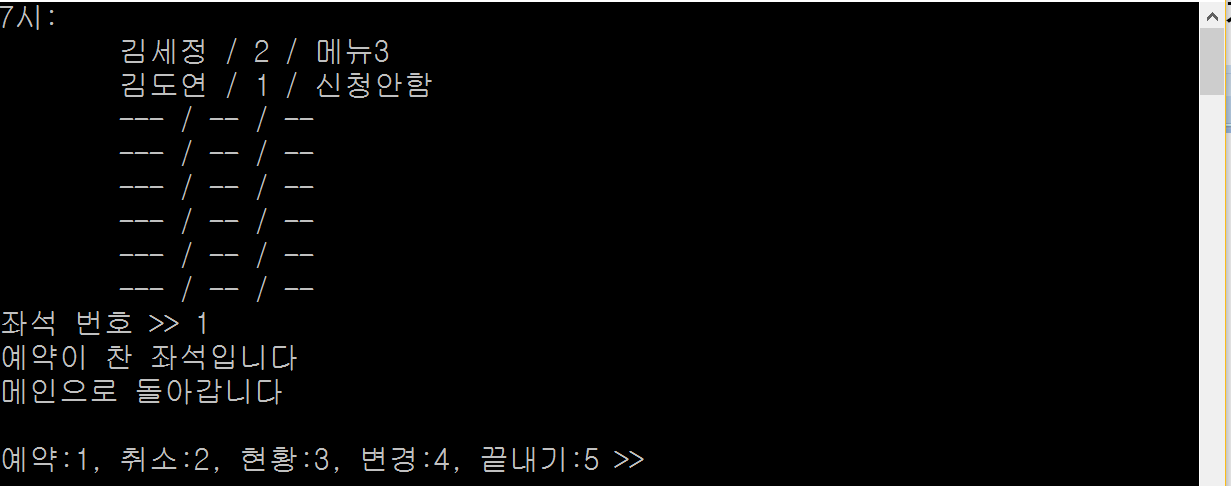
fee = (hydrates \* 20000) + menuprice[food - 1]; //요금은 (신청한 수하물 수 x20000원) 더하기 선택한 기내식메뉴의 가격

}

cout << "최종 추가 요금은 " << fee << "KRW입니다" << endl;

} F. 이미 예약된 좌석을 예약하려는 경우

seat.cpp에 있던 bool함수를 schedule.cpp에서 활용한다.



bool Schedule::booling(int no) { //예약하거나 변경할경우 자리가 겹치는 경우를 구분하기위해 seat.h에 있던 bool함수 isBooked를 사용.

if (seat[no - 1].isBooked() == true)

return false;

else

return true;

}

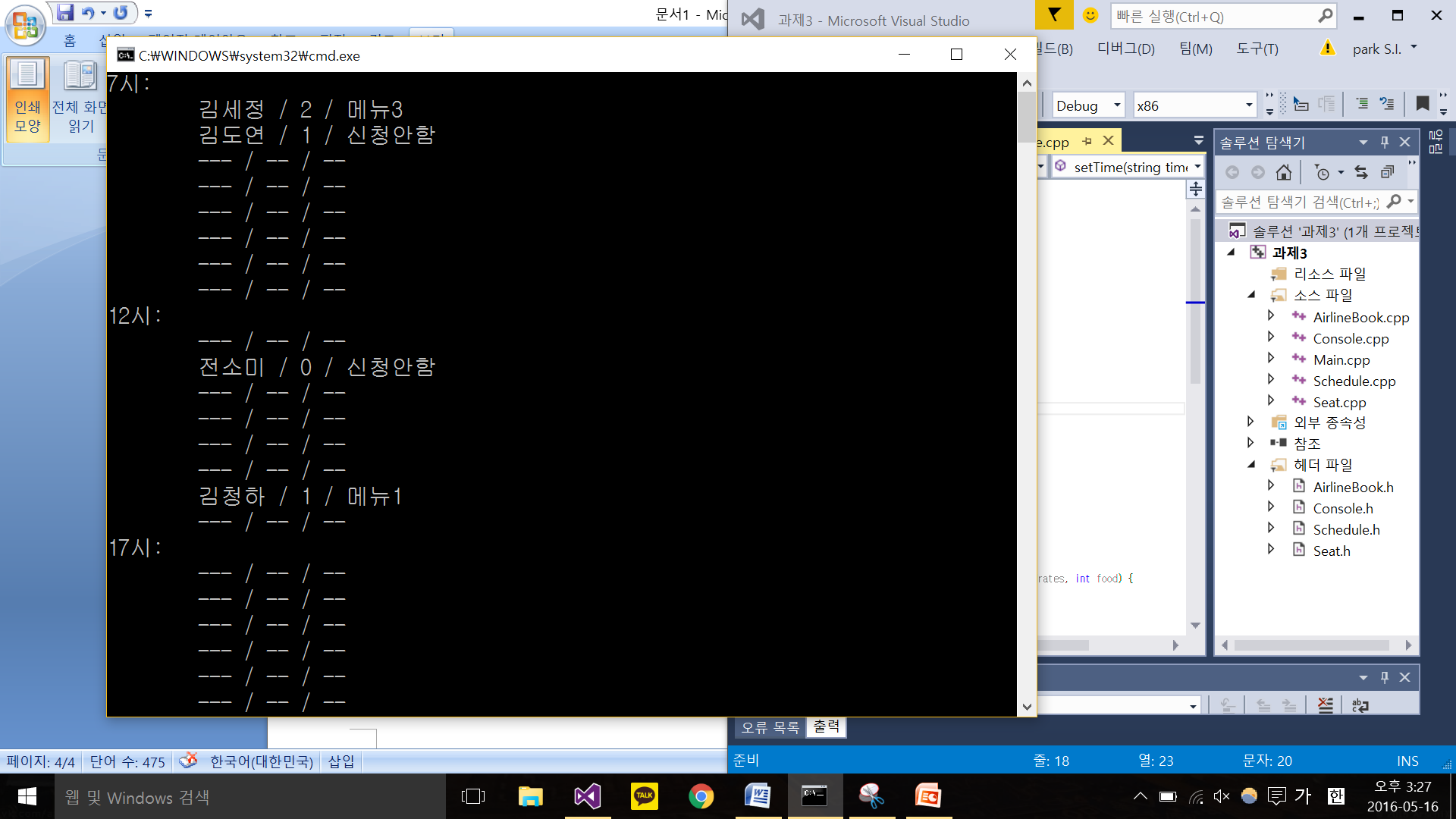
ㄴ. 취소

본래 코드에서 변화없음

ㄷ. 현황

앞서 말했듯이 이름 / 수화물수 / 기내식선택메뉴 로 세로로 출력

캡쳐 화면에 보이는 것은 미리 네 명의 예약정보를 입력한 상태



void Seat::view() {

if (isBooked()) {

if (food == 0) { //기내식 메뉴를 선택하지 않았을 경우

cout << name << " / " << hydrates << " / " << "신청안함" << "\t";

}

else

cout << name << " / " << hydrates << " / " << "메뉴" << food << "\t"; //기내식 메뉴 선택했을경우

}

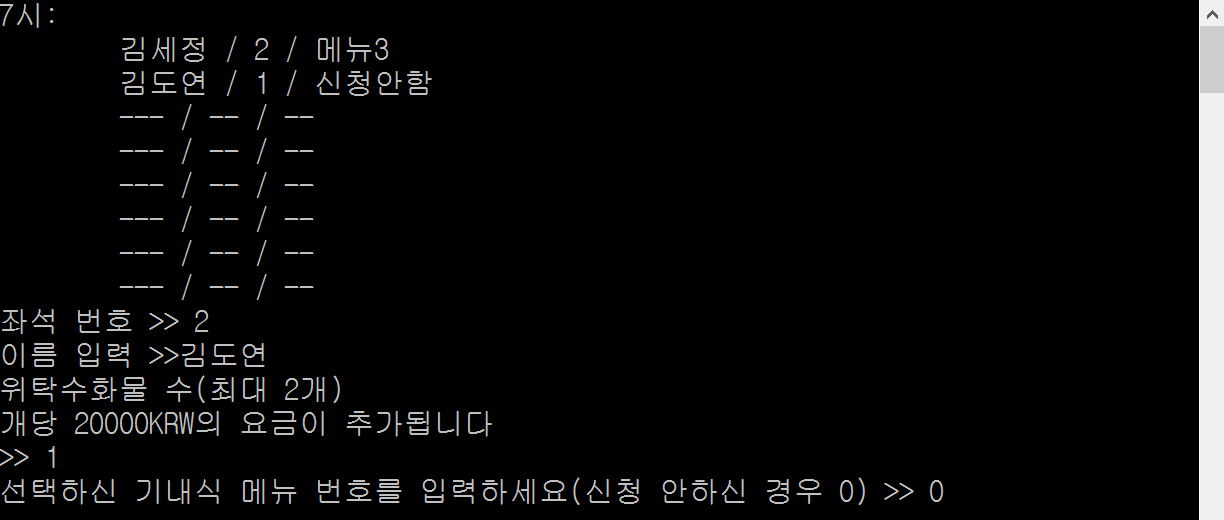
else

cout << "---" << " / " << "--" << " / " << "--" << "\t"; //어떤 입력도 들어가지 않았을 경우

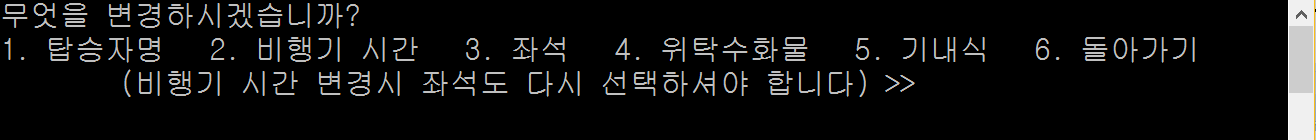
}ㄹ. 변경

예약변경을 위한 메뉴

변경하려는 좌석번호, 이름, 수화물 수, 기내식 선택메뉴를 입력한 후 시작한다.



정보 입력 후 나오는 선택 화면



void AirlineBook::change() { //예약을 변경한다

system("cls"); //화면 지우기

int s, change; //스케쥴 입력을 위한 변수s와 무엇을 변경할지 고르기 위한 변수 change

string bookName;

int seatNo, plusHydrate; //예약한 자리 번호와 수화물갯수를 위한 변수

int meal, meal\_change; //신청했던 기내식메뉴번호와 바꿀 기내식 메뉴번호를 받기위한 변수

s = Console::getScheduleMenu(); // 사용자가 선택한 스케쥴을 입력 받는다.

view(s); // 스케쥴 s의 예약 상황을 출력한다.

seatNo = Console::getSeatNo(); // 좌석 번호를 입력받는다.

if (sche[s - 1].booling(seatNo) == true) { //해당 좌석이 예약되어있지 않은 좌석일 경우

cout << "예약정보가 없습니다" << endl;

}

else { //예약이 된 좌석인 경우

bookName = Console::getName(); // 예약자의 이름을 입력받는다.

plusHydrate = Console::hydrate(); //예약자가 신청한 수화물 수를 받는다

cout << "선택하신 기내식 메뉴 번호를 입력하세요(신청 안하신 경우 0) >> ";

cin >> meal; //예약자가 신청한 기내식 번호를 받는다.

system("cls"); //화면 지우기

cout << "무엇을 변경하시겠습니까?" << endl;

cout << "1. 탑승자명 2. 비행기 시간 3. 좌석 4. 위탁수화물 5. 기내식 6. 돌아가기" << endl;

cout << "\t(비행기 시간 변경시 좌석도 다시 선택하셔야 합니다) >> ";

cin >> change;

switch (change) {

case 1: //탑승자명 변경

cout << "변경하실 탑승자 명을 입력하세요" << endl;

cout << ">> ";

sche[s - 1].cancel(seatNo, bookName); //해당좌석의 예약을 취소

bookName = Console::getName(); //탑승자명 새로 받는다

sche[s - 1].book(seatNo, bookName, plusHydrate, meal); //새로받은 탑승자명으로 다시 예약

view(s); //결과화면을 보인다

break;

case 2: //비행기 시간 변경

cout << "변경하실 비행기 시간을 입력하세요" << endl;

cout << ">> ";

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소

s = Console::getScheduleMenu(); //변경할 비행기 시간을 입력받는다

view(s); //해당 비행기 시간을 보인다

seatNo = Console::getSeatNo(); //해당 비행기 시간에서 앉을 좌석번호를 받는다.

if (sche[s - 1].booling(seatNo) == true) { //만약 좌석이 비어있다면

sche[s - 1].book(seatNo, bookName, plusHydrate, meal);// 해당 스케쥴 예약

view(s); //결과화면을 보인다

}

else

cout << "예약이 찬 좌석입니다" << endl; //예약이 찬 상태라면

break;

case 3: //좌석 변경

cout << "변경하실 좌석번호를 입력하세요" << endl;

cout << ">> ";

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소

seatNo = Console::getSeatNo();

if (sche[s - 1].booling(seatNo) == true) {

sche[s - 1].book(seatNo, bookName, plusHydrate, meal);// 해당 스케쥴 예약

view(s);

}

else

cout << "예약이 찬 좌석입니다" << endl;

break;

case 4: //위탁 수화물 갯수 변경

cout << "변경하실 위탁수화물 수를 입력하세요" << endl;

cout << ">> ";

plusHydrate = Console::hydrate(); //변경할 위탁 수화물 수 입력

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소

sche[s - 1].book(seatNo, bookName, plusHydrate, meal); //변경받은 위탁 수화물 수로 다시 예약

cout << endl;

sche[s - 1].fee(plusHydrate, meal); //바뀐 요금을 계산한다.

break;

case 5: //기내식 변경

cout << "기내식 추가 : 1 기내식 취소 : 2 기내식 변경 : 3" << endl;

cout << " >> ";

cin >> meal\_change;

switch (meal\_change) {

case 1: //기내식 추가

if (meal == 0) { //기내식을 신청하지 않았을 경우

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소한다

meal = Console::food(); //신청할 기내식 메뉴번호를 받는다

sche[s - 1].book(seatNo, bookName, plusHydrate, meal); //변경할 기내식 메뉴번호로 다시 해당좌석을 예약한다.

sche[s - 1].fee(plusHydrate, meal); //바뀐 추가요금을 계산하여 표시한다.

break;

}

else { //기내식을 이미 신청한 경우

cout << "이미 기내식을 신청하셨습니다!" << endl;

break;

}

case 2: //기내식 취소

if (meal == 0) { //기내식을 신청하지 않았을 경우

cout << "기내식을 신청하지 않으셨습니다!" << endl;

break;

}

else { //기내식을 신청했을 경우

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소한다

meal = 0; //기내식번호를 받는 변수에 0 주어준다.

sche[s - 1].book(seatNo, bookName, plusHydrate, meal); //meal값에 0을 넣어 다시 해당좌석 예약한다.

cout << "기내식 신청을 취소하였습니다" << endl;

sche[s - 1].fee(plusHydrate, meal); //바뀐 추가요금을 계산하여 표시한다.

break;

}

case 3: //기내식 변경

if (meal == 0) { //기내식을 신청하지 않았을 경우

cout << "기내식을 신청하지 않으셨습니다!" << endl;

break;

}

else { //기내식을 신청했을 경우

sche[s - 1].cancel(seatNo, bookName); //해당 좌석의 예약을 취소한다

meal = Console::food(); //meal값에 새로운 기내식메뉴 번호를 받는다.

sche[s - 1].book(seatNo, bookName, plusHydrate, meal);

cout << "기내식 변경을 완료하였습니다" << endl;

sche[s - 1].fee(plusHydrate, meal); // 바뀐 추가요금을 계산하여 표시한다.

break;

}

}

break;

case 6: //돌아가기

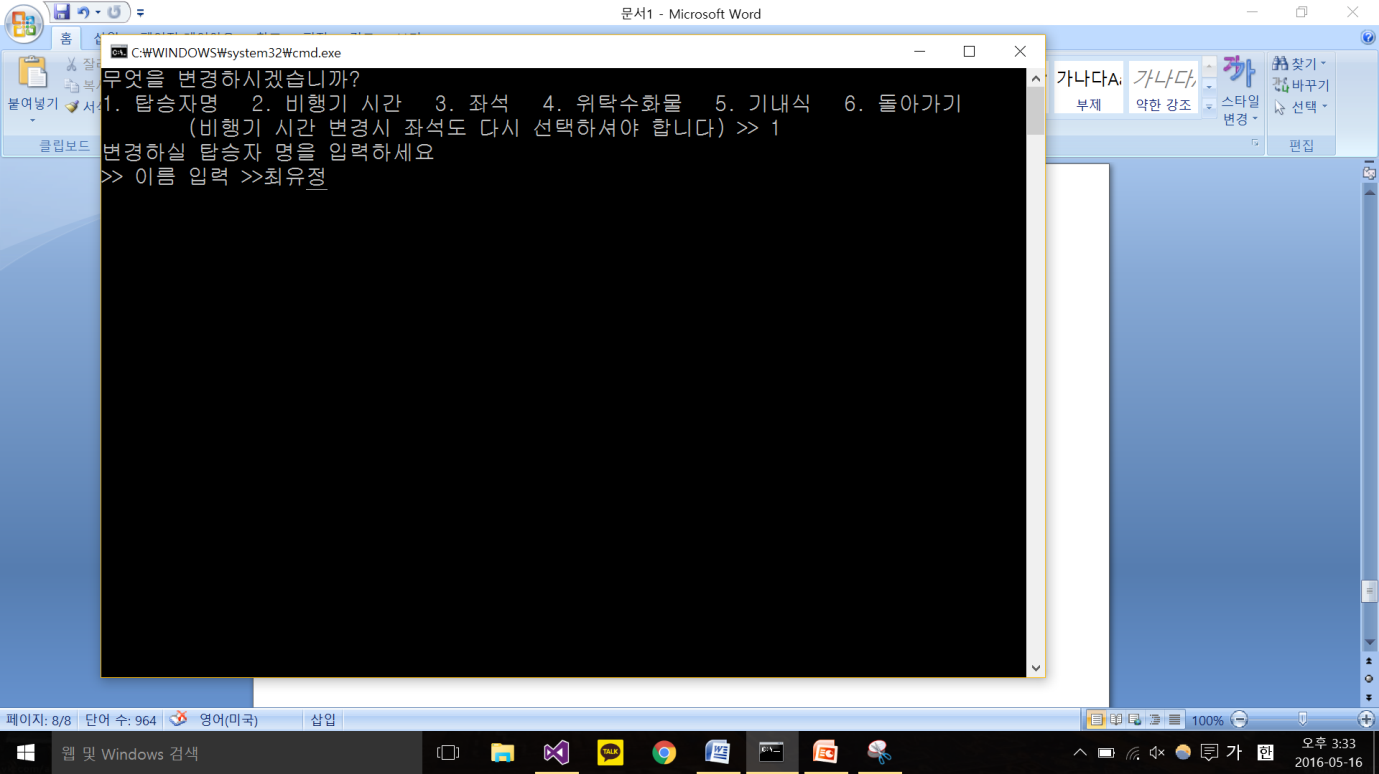
cout << "변경 취소되었습니다" << endl;

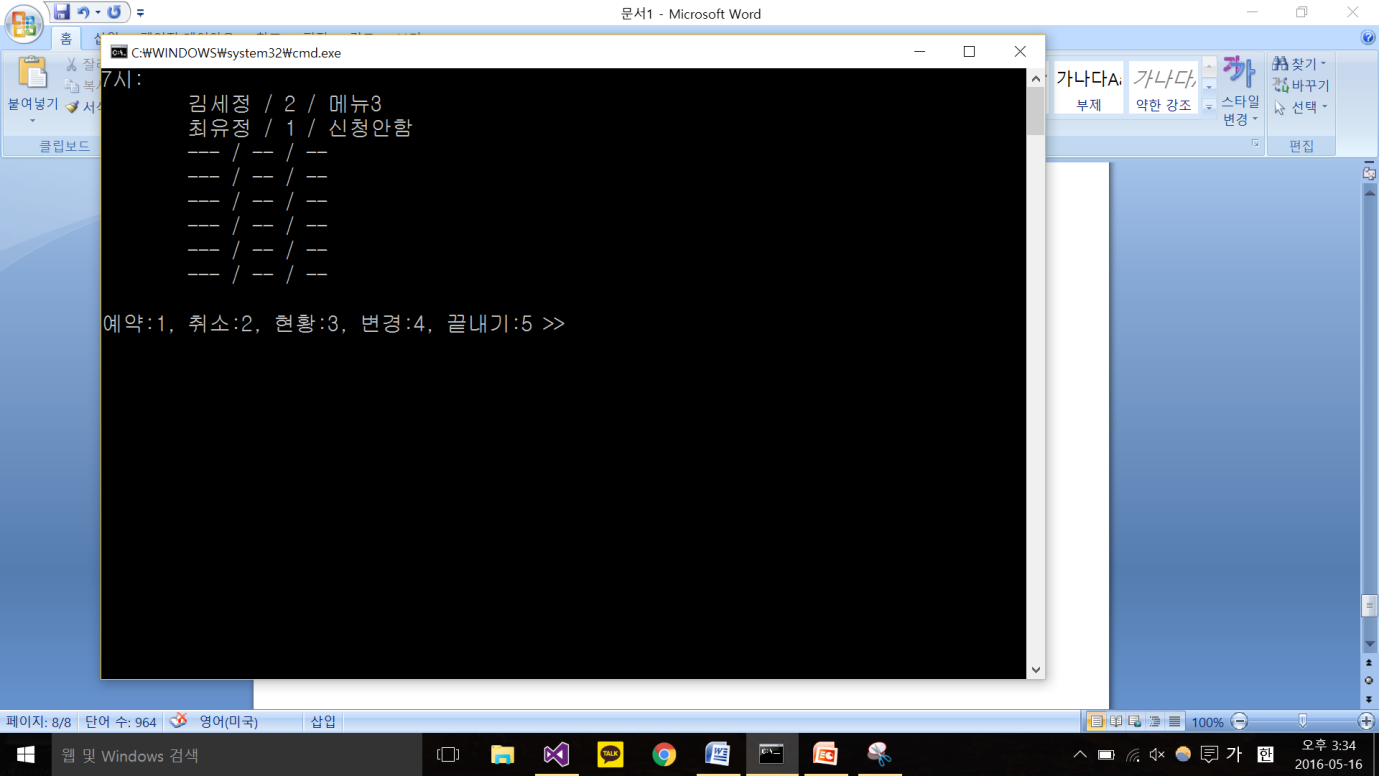
cout << "메인으로 돌아갑니다" << endl;

}

}

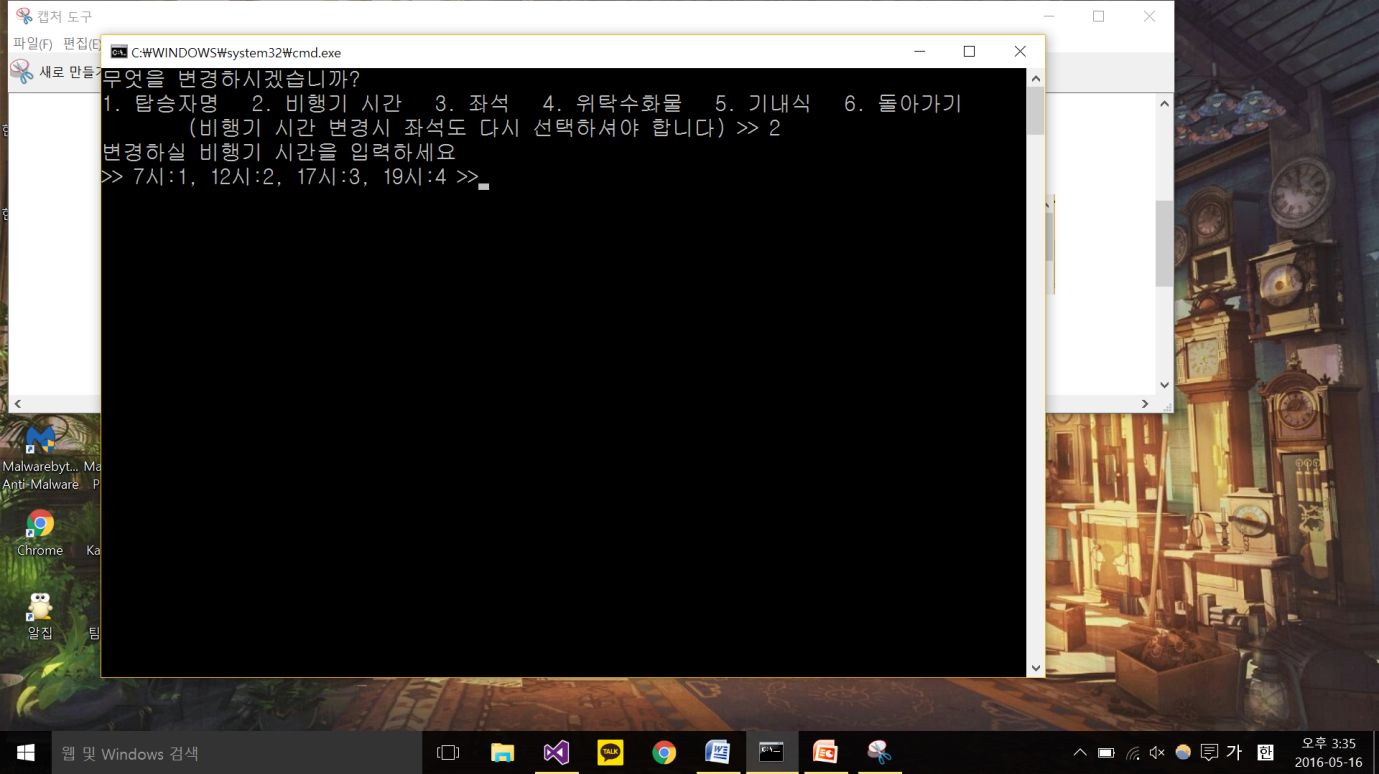
} A. 이름변경

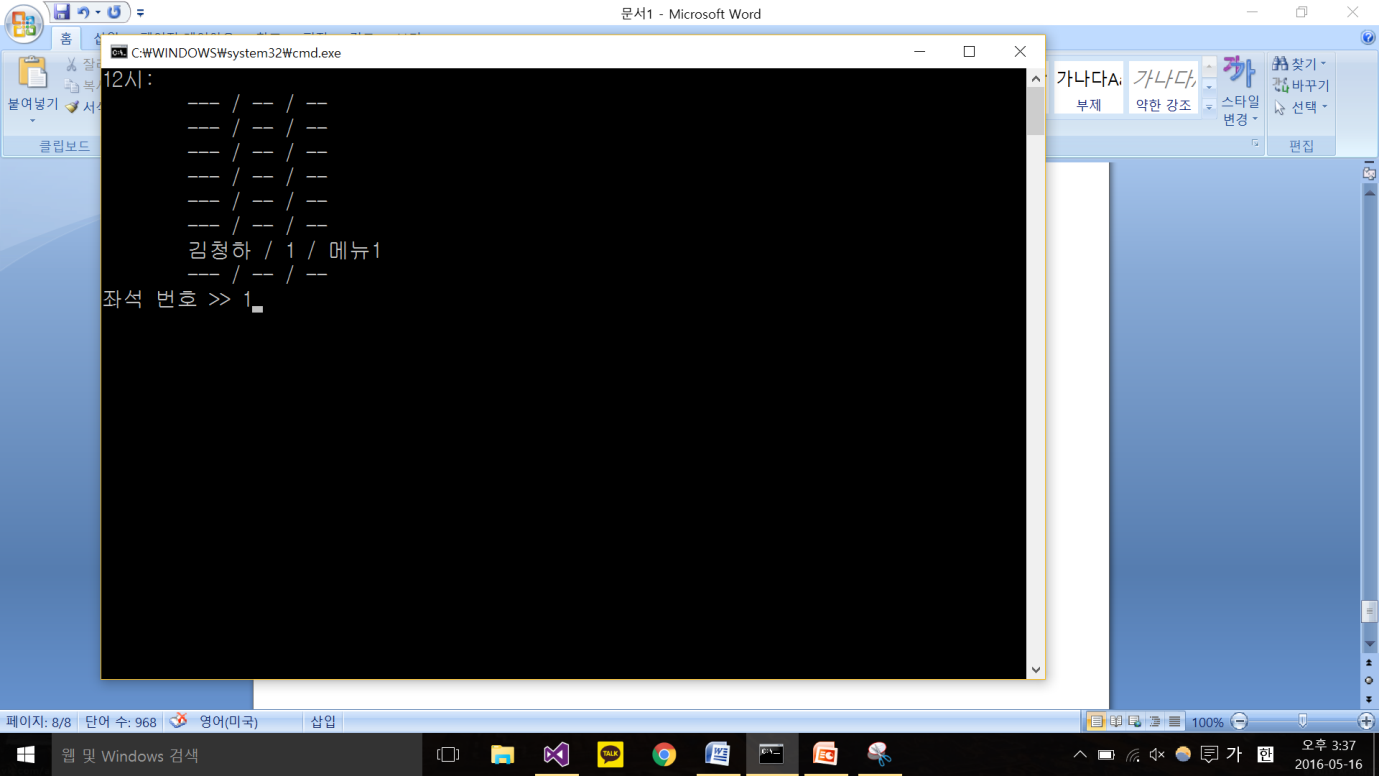




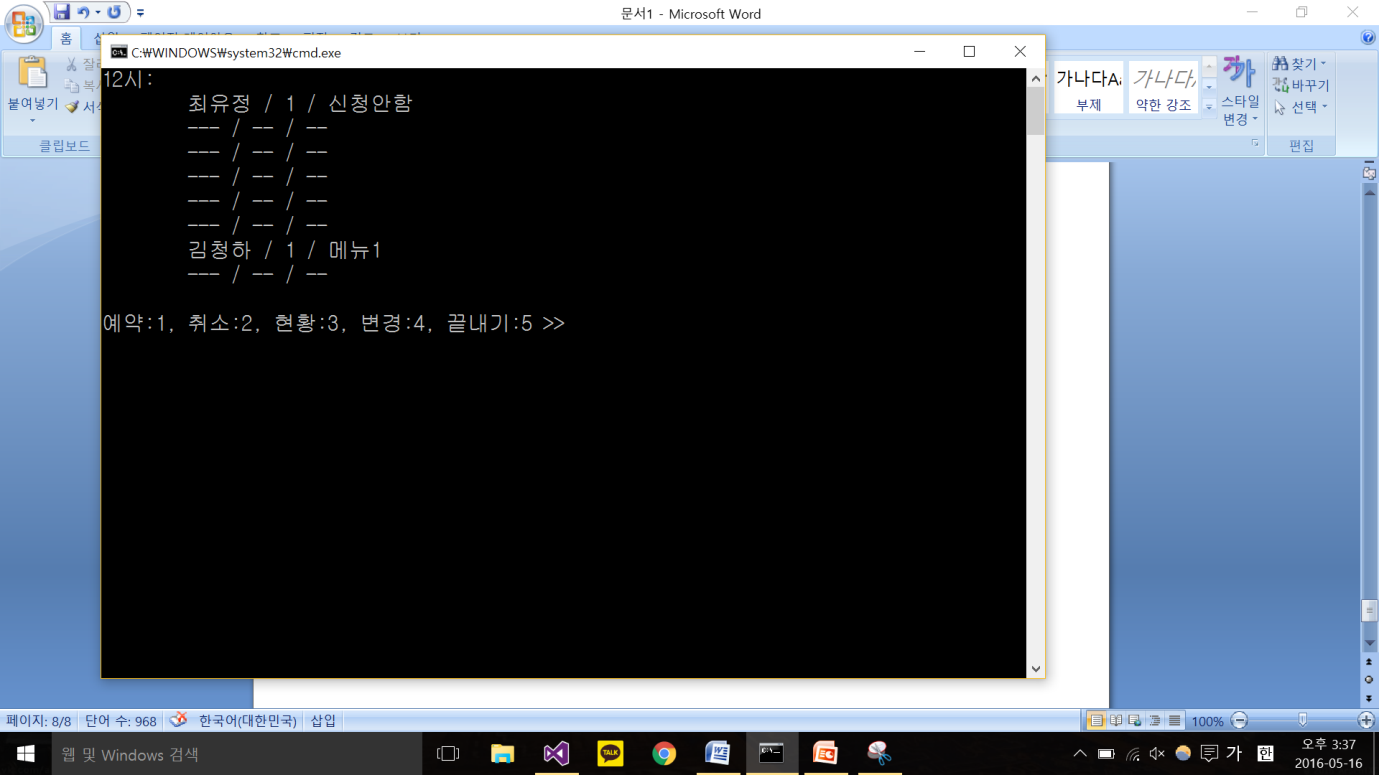
B. 비행기 시간 변경

비행기 시간 변경 시 좌석도 초기화되어 다시 선택 해야한다.

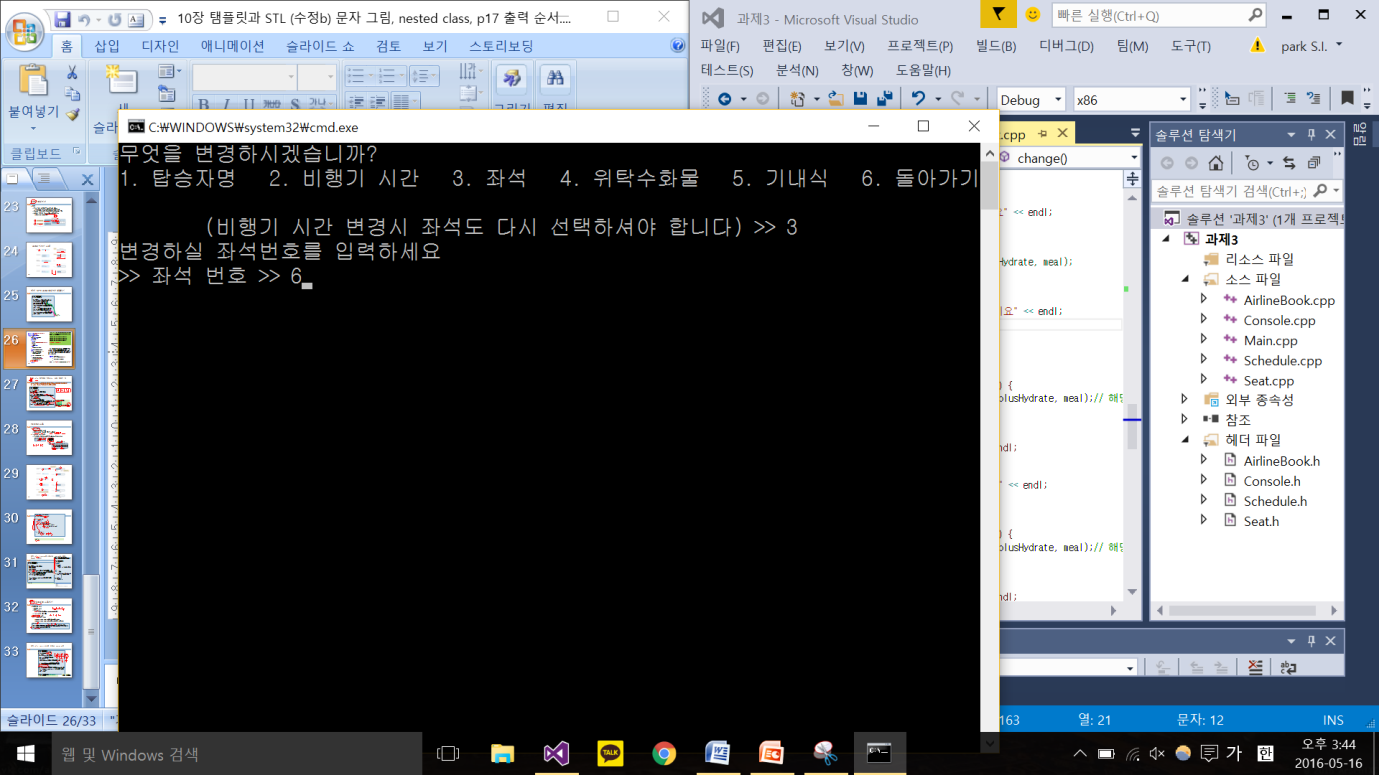


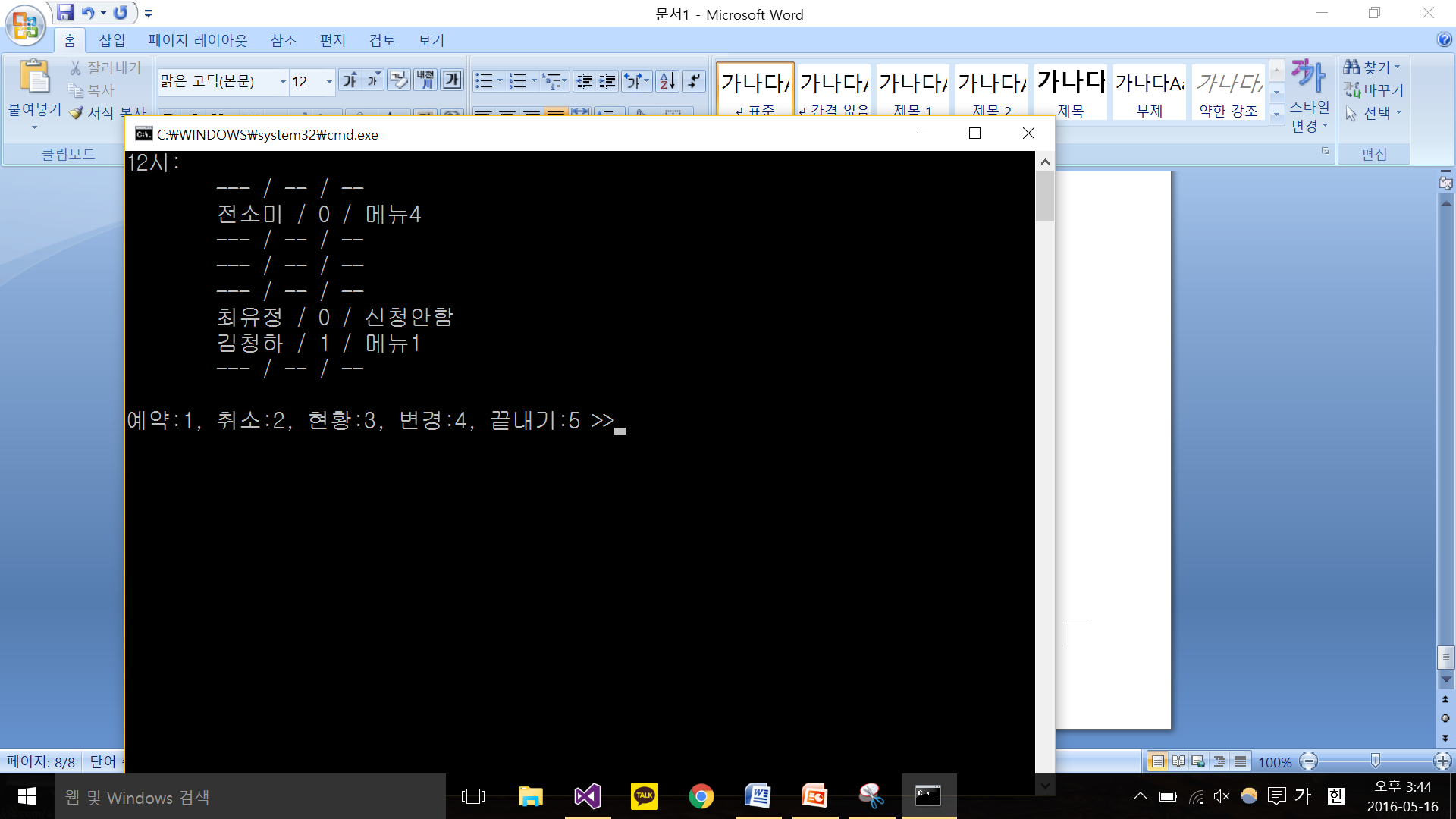


변경된 모습

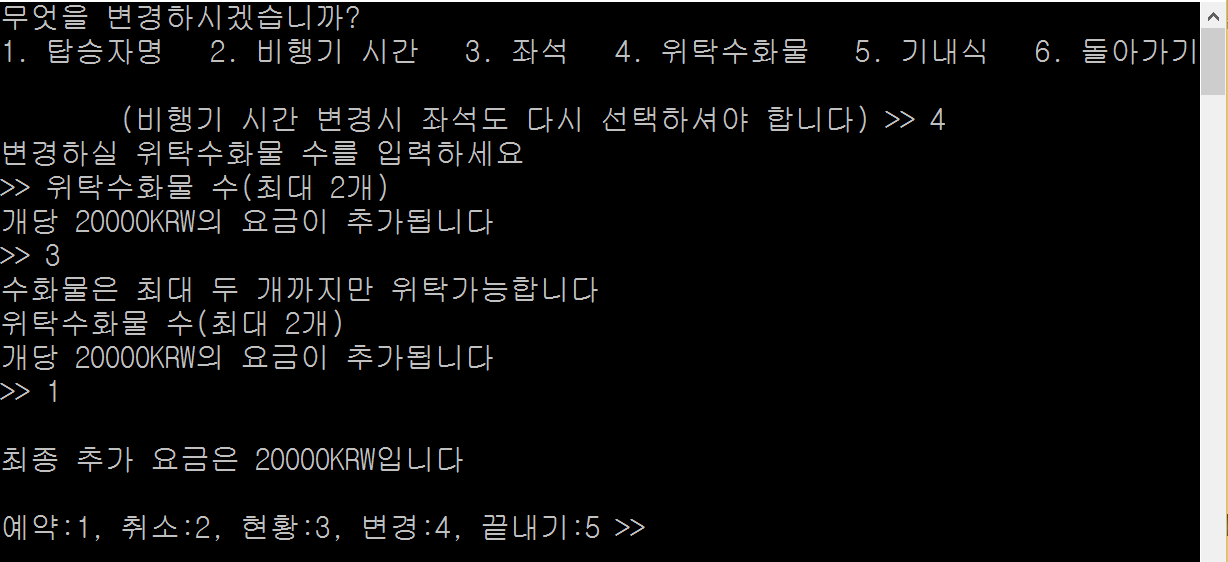


C. 좌석 변경



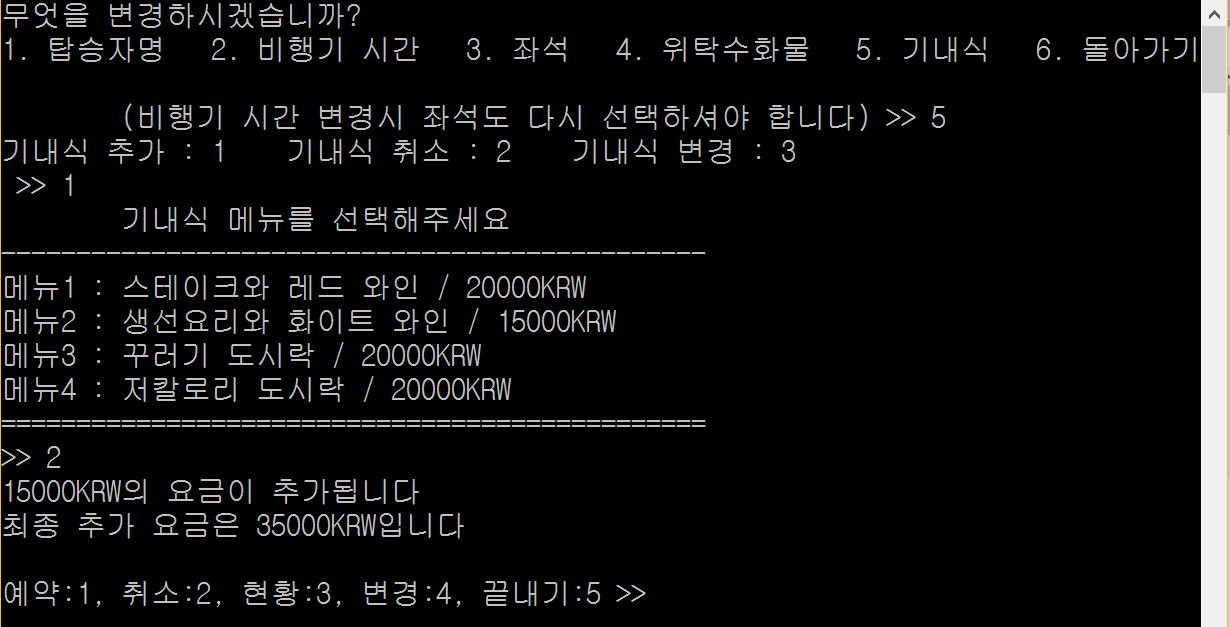


D. 수화물 변경

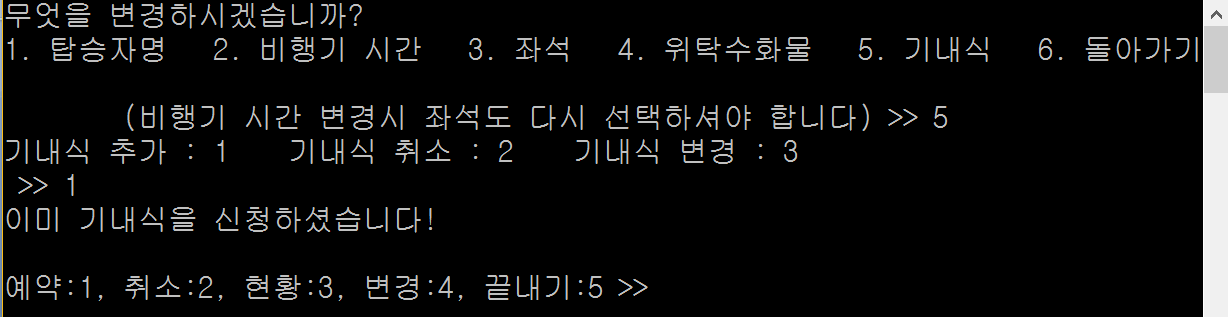


E. 기내식 변경

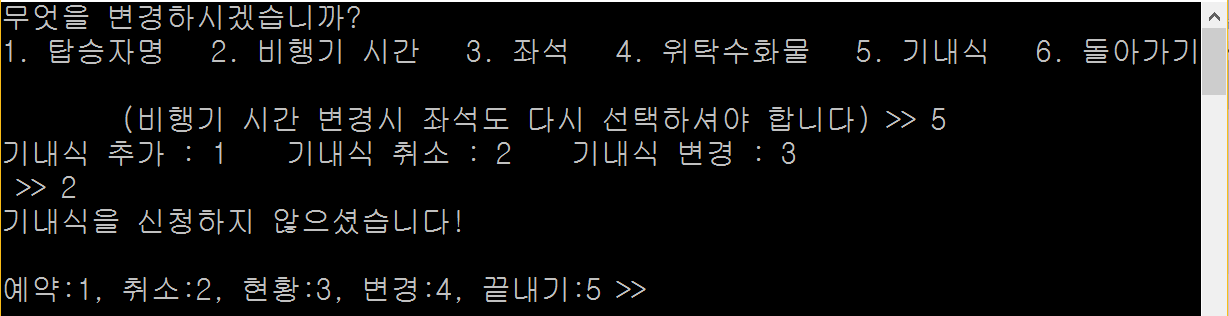
E-1. 기내식 추가(기내식 신청 안 했던 경우)



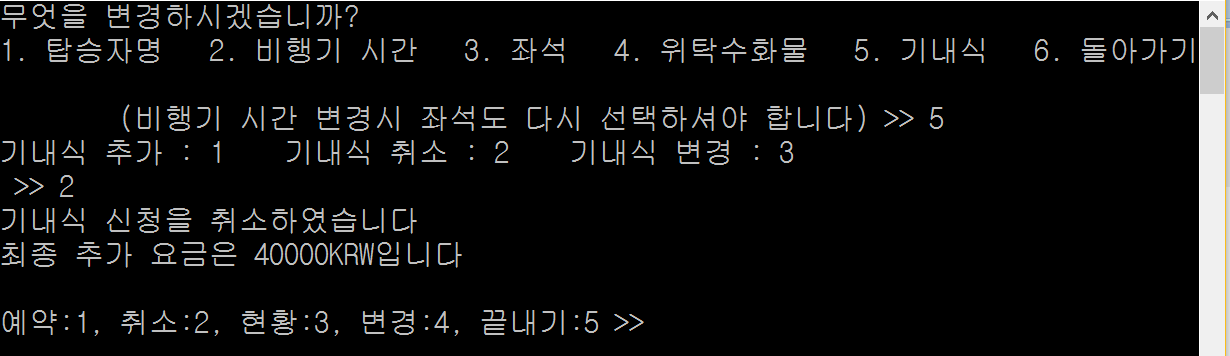
E-1. 기내식 추가(이미 기내식을 추가한 경우)



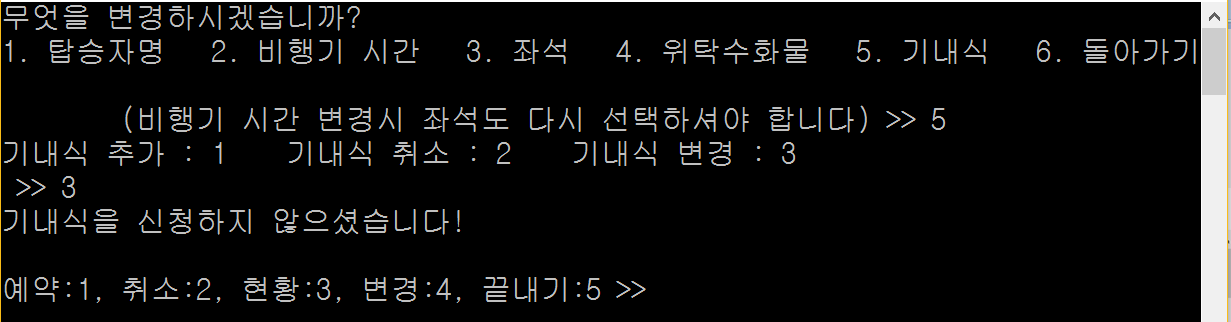
E-2. 기내식 취소 (기내식을 추가하지 않았을 경우)



E-2. 기내식 취소 (기내식을 추가했을 경우)



E-3. 기내식 변경 (기내식을 추가하지 않았을 경우)



E-3. 기내식 변경 (기내식을 추가했을 경우)

