

과제 1번

< 사용 변수 >

char [50] 형 space는 입력한 내용을 저장하는 배열이고, char형 used는 사용된 칸 수를, char 형 count는 입력된 글자를 세는 변수다.

< 풀이 방법 >

[main 함수]

- 먼저 어떤 자료형을 입력받을 것인지를 scanf를 사용해 입력받고 이를 type에 저장한다. switch문을 사용해 type에 따라 다른 scanf 방법을 사용하는데, char 형인 경우, 해당 포인터 값을 (char*)으로 변환하여 입력받고, short형은 (short*)으로, int 형은 (int*)으로, long long 은 (long long*) 형으로 변환하여 입력받아야 한다. 이래야 해당 자료형에 맞는 크기의 char 배열에 입력 값을 저장하게 된다. 물론, 이후 입력을 받기 전 50-used>=1인지를 확인해 입력이 가능한지 판별한다. 만약 불가능한 경우, 저장 공간이 가득 찼음을 알리고 저장된 내용을 print한다. 먼저 입력된 단어수를 출력하고, 여기서도 먼저 나온 type에 따라 형 변환을 한 후 입력값을 출력해야 한다.

< 틀린 이유 분석 >

line 1에서 error LNK1120이 발생했는데, 알고 보니 print 라고 작성해서였다... 물론 이 에러를 해결해도 반복문에 제대로 작동하지 않는 문제가 발생하고, while을 사용해 반복문 문제를 해결해도 제대로 작동하지 않는다. 이는 형 변환으로 인한 것으로, space 배열은 char 형 배열이기에 이 안에 다른 자료 형을 저장하거나 출력하는 경우 이에 따른 적절한 형 변환이 이뤄졌어야 했다. 그러나 이를 제대로 하지 않아 전반적인 문제가 발생한 듯하다.

```
= 50 바이트 남음
> 1 15
= 48 바이트 남음
> 2 10000
= 45 바이트 남음
> 3 1931
= 40 바이트 남음
> 4 9382
= 31 바이트 남음
> 1 15
= 29 바이트 남음
> 2 10000
= 26 바이트 남음
> 3 1931
= 21 바이트 남음
> 4 9382
= 12 바이트 남음
> 1 15
= 10 바이트 남음
> 2 10000
= 7 바이트 남음
> 4 9382
= 저장 공간이 가득 찼습니다.
10
15C 10000S 1931I 9382LL 15C 10000S 1931I 9382LL 15C 10000S

20224382 안선우
```

과제 2번

< 사용 변수 >

- struct CITY에는 이름을 저장할 char [10]형의 name과 주인을 저장할 char 형 owner 멤버 변수가 있다. 해당 자료형의 변수로는 cities[]가 cities는 구조체 배열로 안에 start, seoul, tokyo, sydney, la, paris, hanoi, new delhi, phuket, cairo가 있다.
- struct PLAYER에는 남은 돈을 저장할 int형 money 멤버 변수와 마지막으로 머무른 장소의 index를 저장하는 char형 place 멤버 변수가 존재한다. 해당 자료형의 변수로는 p1과 p2가 있으며, 각가 플레이어 1과 2를 상징한다. p1과 p2는 각자 다른 구조체 변수이다.
- round, dice, flag은 모두 char 형 자료형 변수로, main 함수 내에서 라운드와 주사위 결과, 그리고 게임 종료 여부를 저장한다.

< 풀이 방법 >

[main 함수]

- while 함수를 통해 최대 30판까지만 게임이 진행되도록 한다. 매 라운드 마다 rand함수를 사용해 나온 주사위 값을 dice에 저장하고, turn함수를 실행한 후, turn 함수에서 반환한 flag 값으로 잔고부족으로 게임을 끝낼지(flag==5인 경우). 아니면 타 플레이어에게 통행세를 내야 하는지(flag==4인 경우)를 판단한다.
- 두 플레이어 모두 잔고부족이 발생하지 않고 게임이 30 라운드 진행된 후에는 실제 '모두의 마블' 규칙에 따라 더 많은 잔고가 남은 사람이 승리하도록 한다.

[turn 함수]

- main 함수에서 현재 플레이하는 플레이어 구조체 변수의 포인터 값, cities[] 구조체 변수의 포인터 값, 주사위 값, 그리고 플레이어 넘버(1 or 2)를 전달받는다. 먼저 player 변수의 place 멤버변수에 dice를 더했을 때 어디에 위치하게 되는지를 dice 변수와 cities[] 변수를 활용해 파악한다. 만일 start 포인트인 cities[0] 혹은 자신이 소유한 도시에 도착한 경우, 잔고만 표시하고 턴을 다음 플레이어에게 넘긴다. 타인 소유 도시에 도착한 경우, 통행세를 낼 수 있는 경우 (*player).money에서 600원을 제하고, 4를 반환해 main 함수에서 다른 플레이어가 통행세를 받을 수 있도록 했다. 만일 통행세를 낼 수 없는 경우, 잔고부족임을 출력하고 5를 반환해 main 함수에서 게임을 종료하도록 한다. 아무도 소유하고 있지 않은 빈 도시에 도착한 경우, 잔고가 남은 경우, (*player).money에서 300원을 제하고 해당 도시를 구입한다. 구입 여부는 cities[(player).place].name에 저장해(1 or 2) 추후 라운드에서 이와 info를 비교해 자신이 소유한 도시인지 아닌지 구분이 가능하도록 한다. 빈 도시에 들어갔으나 구입할 돈이 없는 경우에는 잔고부족만 출력하고 다음 플레이어에게 턴을 넘긴다.

```
30번째 라운드
player 1의 주사위: 5
Cairo (1)
player 1의 잔고 6500

player 2의 주사위: 6
Hanoi (1)
통행세 600원을 납부합니다.
player 2의 잔고 200

player 1의 잔고가 더 많습니다. player 1이 승리했습니다.
```

< 30번째 라운드까지 진행했으나 둘 다 잔고가 남은 경우, 더 많이 남긴 사람이 승리함.

√ 괄호 안의 수는 라운드임
ex. (2) = 2번째 라운드

결과					
1번째 라운드 player 1의 주사위: 2 Tokyo (주인없음) player 1가 Tokyo를 구입합니다. player 1의 잔고 4700 player 2의 주사위: 1 Seoul (주인없음) player 2가 Seoul를 구입합니다. player 2의 잔고 4700 2번째 라운드 player 1의 주사위: 2 LA (주인없음) player 1가 LA를 구입합니다. player 1의 잔고 4400 player 2의 주사위: 3 LA (1) 통행세 600원을 납부합니다. player 2의 잔고 4100 3번째 라운드 player 1의 주사위: 2 Hanoi (주인없음) player 1가 Hanoi를 구입합니다. player 1의 잔고 4700 player 2의 주사위: 6 Start (시작포인트) player 2의 잔고 4100 4번째 라운드 player 1의 주사위: 6 Tokyo (1) player 1의 잔고 4700 player 2의 주사위: 6 Hanoi (1) 통행세 600원을 납부합니다. player 2의 잔고 3500	5번째 라운드 player 1의 주사위: 6 Phuket (주인없음) player 1가 Phuket를 구입합니다. player 1의 잔고 5000 player 2의 주사위: 2 Phuket (1) 통행세 600원을 납부합니다. player 2의 잔고 2900 6번째 라운드 player 1의 주사위: 6 LA (1) player 1의 잔고 5600 player 2의 주사위: 3 Seoul (2) player 2의 잔고 2900 7번째 라운드 player 1의 주사위: 4 Phuket (1) player 1의 잔고 5600 player 2의 주사위: 4 Paris (주인없음) player 2가 Paris를 구입합니다. player 2의 잔고 2600 8번째 라운드 player 1의 주사위: 5 Sydney (주인없음) player 1가 Sydney를 구입합니다. player 1의 잔고 5300 player 2의 주사위: 5 Start (시작포인트) player 2의 잔고 2600	9번째 라운드 player 1의 주사위: 1 LA (1) player 1의 잔고 5300 player 2의 주사위: 4 LA (1) 통행세 600원을 납부합니다. player 2의 잔고 2000 10번째 라운드 player 1의 주사위: 3 New Delhi (주인없음) player 1가 New Delhi를 구입합니다. player 1의 잔고 5600 player 2의 주사위: 1 Paris (2) player 2의 잔고 2000 11번째 라운드 player 1의 주사위: 2 Cairo (주인없음) player 1가 Cairo를 구입합니다. player 1의 잔고 5300 player 2의 주사위: 5 Start (시작포인트) player 2의 잔고 2000 12번째 라운드 player 1의 주사위: 2 Seoul (2) 통행세 600원을 납부합니다. player 1의 잔고 4700 player 2의 주사위: 1 Seoul (2) player 2의 잔고 2600	13번째 라운드 player 1의 주사위: 6 New Delhi (1) player 1의 잔고 4700 player 2의 주사위: 5 Hanoi (1) 통행세 600원을 납부합니다. player 2의 잔고 2000 14번째 라운드 player 1의 주사위: 1 Phuket (1) player 1의 잔고 5300 player 2의 주사위: 6 Tokyo (1) 통행세 600원을 납부합니다. player 2의 잔고 1400 15번째 라운드 player 1의 주사위: 1 Cairo (1) player 1의 잔고 5900 player 2의 주사위: 2 LA (1) 통행세 600원을 납부합니다. player 2의 잔고 800 16번째 라운드 player 1의 주사위: 4 Sydney (1) player 1의 잔고 6500 player 2의 주사위: 1 Paris (2) player 2의 잔고 800	17번째 라운드 player 1의 주사위: 5 Phuket (1) player 1의 잔고 6500 player 2의 주사위: 3 Phuket (1) 통행세 600원을 납부합니다. player 2의 잔고 200 18번째 라운드 player 1의 주사위: 4 Tokyo (1) player 1의 잔고 7100 player 2의 주사위: 5 Sydney (1) 통행세를 납부할 수 없습니다. (원인: 잔고부족) player 2의 잔고 200 player 1이 승리했습니다. 20224382 안선우	

0 start	1 seoul	2 tokyo	3 sydney	4 la	5 paris	6 hanoi	7 new delhi	8 phuket	9 cairo
<p>통행(3) p2</p> <p>통행(8) p2</p> <p>통행(11) p2</p>	<p>구입(1) p2</p> <p>통행(6) p2</p> <p>통행(12) p1</p> <p>통행(12) p2</p>	<p>구입(1) p1</p> <p>통행(4) p1</p> <p>통행(14) p2</p> <p>통행(18) p1</p>	<p>구입(8) p1</p> <p>통행(16) p1</p> <p>통행(18) p1</p>	<p>구입(2) p1</p> <p>통행(2) p2</p> <p>통행(6) p1</p> <p>통행(9) p1</p> <p>통행(9) p2</p> <p>통행(15) p2</p>	<p>구입(7) p2</p> <p>통행(10) p2</p> <p>통행(16) p2</p>	<p>구입(3) p1</p> <p>통행(4) p2</p> <p>통행(13) p2</p>	<p>구입(10) p1</p> <p>통행(13) p1</p>	<p>구입(5) p1</p> <p>통행(5) p2</p> <p>통행(7) p1</p> <p>통행(14) p1</p> <p>통행(17) p1</p> <p>통행(17) p2</p>	<p>구입(11) p1</p> <p>통행(15) p1</p>
	<p>p1 -600</p> <p>p2 +300</p>	<p>p1 +300</p> <p>p2 -600</p>	<p>p1 -300</p> <p>p2</p>	<p>p1 +1500</p> <p>p2 -1800</p>	<p>p1</p> <p>p2 -300</p>	<p>p1 +900</p> <p>p2 -1200</p>	<p>p1 -300</p> <p>p2</p>	<p>p1 +900</p> <p>p2 -1200</p>	<p>p1 -300</p> <p>p2</p>

∴ p1 잔고 : 7100 // p2 잔고 : 200

과제 3번

< 사용 변수 >

- struct BOOK 구조체는 책 정보를 저장하는 구조체이다. char[50]형의 title, char[30]형의 author, char[20]형의 publisher, char형의 current 멤버변수가 존재한다. 순서대로 책 제목, 저자, 출판사, 대출/보유 여부를 나타낸다. current가 0이면 도서관 보유 중, 1이면 대출 중을 의미한다. main 함수에서 books[100]가 BOOK 구조체 변수이다.
- struct USER 구조체는 회원 정보를 저장하는 구조체이다. char[30]형의 name, char[15]형의 phone 멤버 변수가 존재한다. 순서대로 회원 명, 회원 전화번호이다. main함수 안에서 users[100]가 USER 구조체 변수로 사용된다.
- struct BORROW 구조체는 대출/반납 기록을 저장하는 구조체이다. struct BOOK* 형의 book 포인터 변수와, struct USER* 형의 borrower 포인터 변수, char[10]형의 borrowdate, char[10]의 returnornot 멤버변수가 존재한다. 순서대로 대출 도서 명, 대출자 회원명, 빌린 날짜, 반납 날짜를 의미한다. main 함수 안에서 history[100]가 BORROW 구조체 변수이다.
- main 함수에서 char 자료형의 work, bookcount, usercount, historycount는 순서대로 작업 종류, 보유 중인 도서 개수, 도서관 회원 수, 기록 수를 의미한다. char[50]형의 temp는 잠시 fgets 정보를 저장해 두는 함수하고, struct BOOK*, struct USER*, struct BORROW* 자료형의 ptr1, ptr2, ptr3 포인터 변수는 이후 history[100] 변수 저장 및 출력 시 사용하는 함수로, NULL 값을 기본으로 한다.

< 풀이 방법 >

[main 함수]

- 기본적으로 5권의 도서와 5명의 회원을 저장해 두었다. while문을 사용해 원하는 업무를 고를 수 있도록 한 후(type에 저장), switch문을 사용해 업무를 진행한다. 도서 대출을 원하는 경우, fgets를 사용해 사용자가 대출하고자 하는 책 이름을 받고, for문과 strcmp를 사용해 books[100].title에 같은 값이 존재하는지 확인한다. 있는 경우, 해당 포인터 값을 ptr1에 저장하고, ptr1->current가 0인지 확인해 대출 가능여부를 확인한다. 이후, 사용자에게 fgets를 통해 빌릴 회원의 이름을 입력받고, 동일하게 for 문과 strcmp를 사용해 users[100].name에 동일한 이름이 있는지 확인한다. 있다면 해당 포인터 값을 ptr2에 저장한다. 이후 대출 날짜를 입력해 history[historycount].borrowdate에 저장하고, .returnornot에는 strcpy를 사용해 "대출중"을 저장한다. 이는, 대출 현황 출력 시 반납이 이뤄지지 않았음을 표현하기 위함이다. ptr1.current를 1로 수정해 대출이 이뤄졌음을 저장하고, ptr1을 history[historycount].book에, ptr2를 .borrower에 저장해 저장공간을 효율적으로 사용하고자 했다.
- 반납의 경우, fgets로 입력받은 책 제목이 books[100]에 존재하는지 확인하고, 있다면 해당 포인터 값을 다시 ptr1에 저장한 후, ptr1->current가 1인지를 확인해 실제로 대출한 책을 반납하는 것인지 확인한다. 해당 책에 대한 내용이 담긴 history[100]의 포인터 값을 ptr3에 저장하고, ptr1->current를 0으로 바꾸고, ptr3->returnornot에 반납 날짜를 저장한다.
- 도서 보유 현황은 for문과 bookcount를 사용해 books[100]을 출력한다. if문을 사용해 book[n].current가 1이면 (대출중)으로, 0이면 (보유중)으로 출력한다.
- 대출 현황은 for 문과 historycount를 사용해 history[100]을 출력한다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황
1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

한비자

** 잘못된 책 제목입니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황
1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

노인과 바다

- 빌릴 회원 이름을 입력하세요.

아아아

** 잘못된 회원명입니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황
1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

노

** 잘못된 책 제목입니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황
1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

노인과 바다

- 빌릴 회원 이름을 입력하세요.

가가가

- 오늘 날짜를 입력하세요. 예)20231202

20231128

대출이 완료되었습니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황
1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

노인과 바다

** 대출 중인 책입니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황

1

1번 '도서 대출'을 누르셨습니다.

- 대출 할 책 제목을 입력하세요.

피로사회

- 빌릴 회원 이름을 입력하세요.

나나나

- 오늘 날짜를 입력하세요. 예)20231202

20231201

대출이 완료되었습니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황

2

2번 '도서 반납'을 누르셨습니다.

- 반납 할 책 제목을 입력하세요.

걸리버 여행기

** 대출하지 않은 책입니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황

2

2번 '도서 반납'을 누르셨습니다.

- 반납 할 책 제목을 입력하세요.

노인과 바다

- 오늘 날짜를 입력하세요. 예)20231202

20231202

반납이 완료되었습니다.

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황

3

3번 '도서 보유 현황'을 누르셨습니다.

1) (클라우드 슈밥의) 제4차 산업혁명 // Schwab, Klaus // 새로운현재 // (보유중)

2) 걸리버 여행기 // Swift, Jonathan // 문예출판사 // (보유중)

3) 노인과 바다 // Hemingway, Ernest // 문학동네 // (보유중)

4) 마음 // 하목수석 // 문예출판사 // (보유중)

5) 피로사회 // 한병철 // 문학과지성사 // (대출중)

원하는 업무를 고르세요.

1: 도서 대출 / 2: 도서 반납 / 3: 도서 보유 현황 / 4: 대출 현황

4

4번 '대출 현황'을 누르셨습니다.

1) 노인과 바다 // 가가가(010-1111-2222) // 20231128 // 20231202

2) 피로사회 // 나나나(010-2222-3333) // 20231201 // 대출중