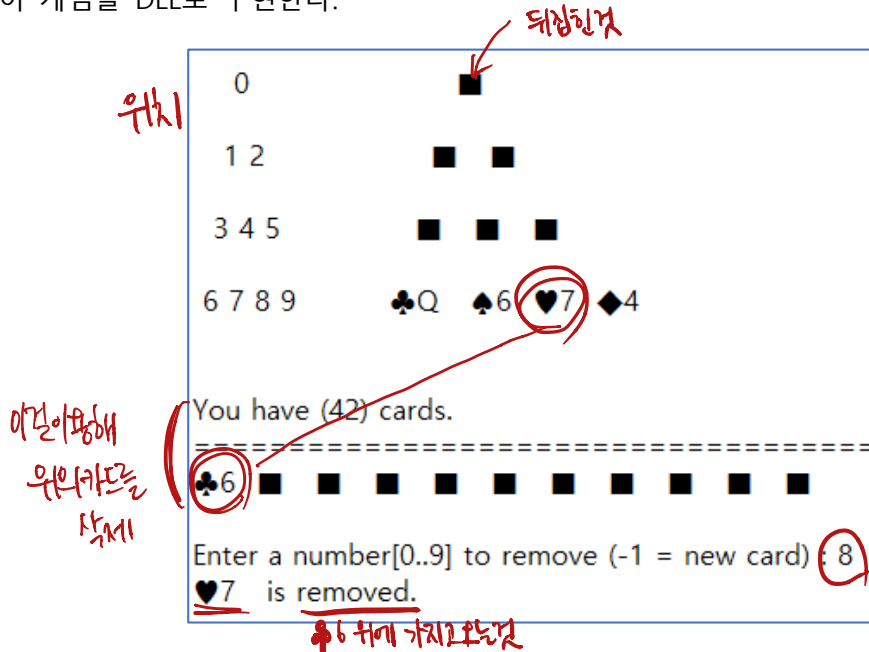


tri-peak?

## 자료구조 프로그래밍 과제 (Solitaire)

솔리테어 게임을 DLL로 구현한다.



(게임 시작)

총 10개의 카드가 위와 같이 나열되어 있다. 가장 하단의 4장의 카드만 오픈되어 있고 나머지 카드는 뒤집어져 있다. 모든 카드를 제거하면 승리하는 게임이다.

(카드제거)

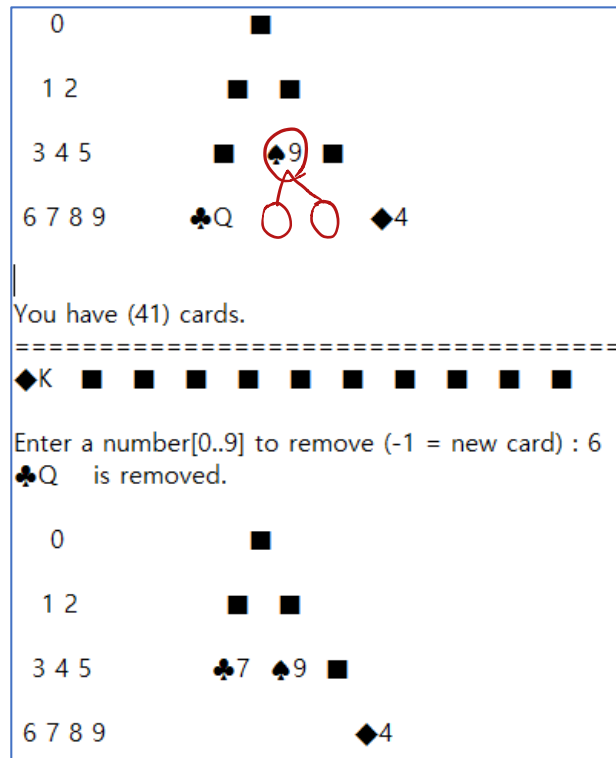
솔리테어는 자신이 가지고 있는 카드와 숫자가 1차이가 나는 카드를 없앨 수 있다.

- 제거할 수 있는 카드 쌍의 예 : (6, 7), (J, 10), (K, A), 등 // A = 1, J=11, Q=12, K=13 (순환구조)
- 왼쪽 상단의 숫자 트리(0~9) 중 하나의 값을 입력(예, 8)하여 제거할 카드 번호를 지정한다.
- 사용자 손에 총 42장의 카드가 있다고 되어 있고, 그 중 가장 위에 있는 카드가 플레이 할 수 있는 스페이드 6번 카드이다. 카드 (스페이드 6)은 트리 8번에 있는 카드 (하트 7)과 1차이가 남으로 제거할 수 있다. 제거되었다는 메시지가 나오는 것을 볼 수 있다.

(게임 패배)

화면의 카드가 모두 제거되기 전에 자신의 카드가 모두 없어지면 게임에서 진다.

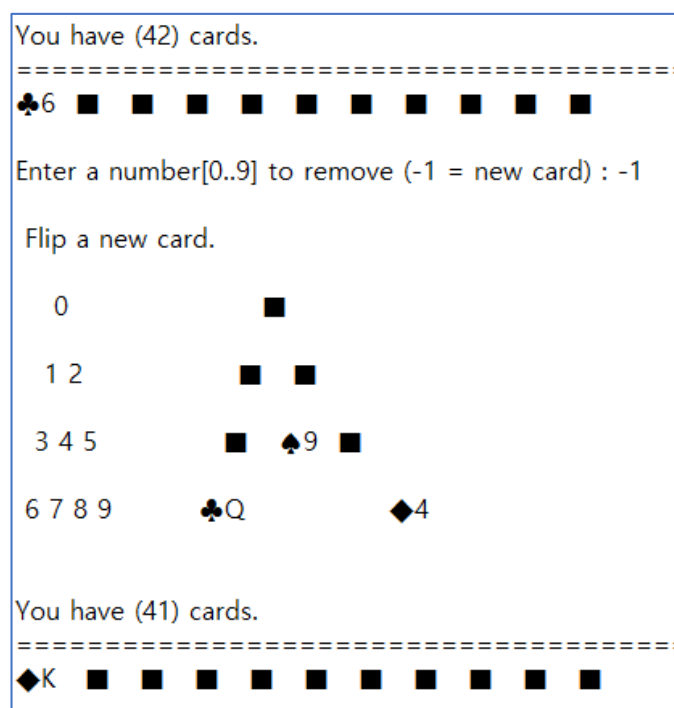
자신의 하위 카드가 모두 제거되면 상위 카드가 오픈된다.



(새 카드 받기)

손에 들고 있는 카드(예, 스페이드 6)로 제거할 카드가 없을 경우, (-1)을 입력하여 새로운 카드를 열 수 있다.

제거된 카드는 자신이 가져오게 되는데 스택 구조가 가장 상단에 올라가서 바로 다음에 사용할 카드가 된다.



(최종 승리)

10장의 카드를 모두 제거하면 게임에서 승리한다.

```
Enter a number[0..9] to remove (-1 = new card) : 2
♠K is removed.

  0          ♦Q
1 2
3 4 5
6 7 8 9

You have (20) cards.
=====
♠K ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter a number[0..9] to remove (-1 = new card) : 0
♦Q is removed.

You Win.
```

<구현 조건>

// Card tree : doubly linked list (tree node is created by malloc())  
// Hand card : linked stack (top node card is to be played)

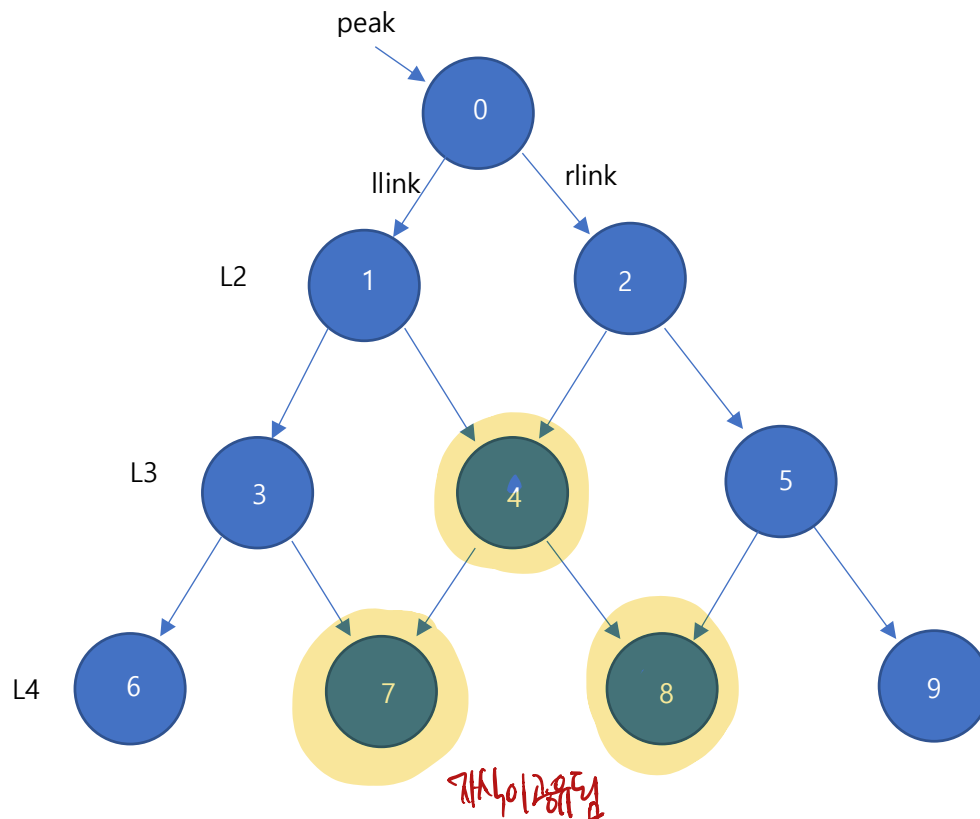
```
struct node_type {
    int num;           // A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
    int shape;         // 0, 1, 2, 3 = {space, heart, clover, diamond} = 2 or 4 sets
    int show;          // 0 = hide, 1 = show
    node_ptr llink, rlink;
};
```

```
int mark[4][13]; //randomly pick cards
```

// pick a card (총 26장(무늬 2세트)일 경우 게임 진행이 원활함)

```
sh = rand() % NUM_SHAPE; // NUM_SHAPE = 2 or 4, (2 is recommended.)
no = rand() % 13;        // 0 = A, 12 = K
```

종북에서받지않게위함.



재시이름  
명함상영장리노트

```
void main()
{
    init_card();
    reset_board();

    while (1)
    {
        show_board();
        if (no match)
        {
            pop_handcard(); // if no more card to play(empty stack), you lose.
        }
        if (all cards removed)
        {
            printf("Wn You Win. Wn");
        }
    }
}
```