

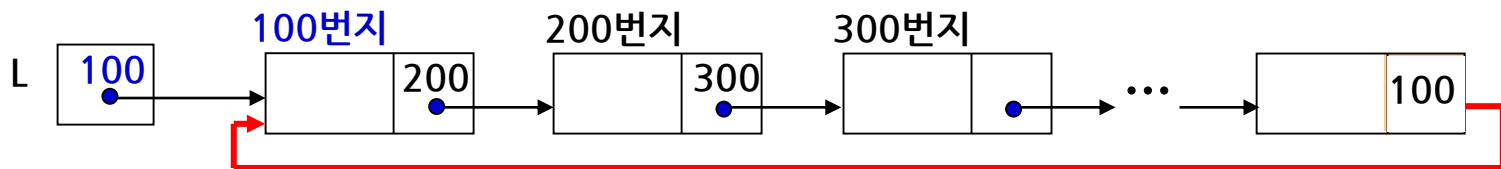
1 | 원형 연결 리스트 특징

1 원형 연결 리스트의 개념

- ▶ 단순 연결 리스트에서 마지막 노드가 리스트의 첫 번째 노드를 가리키게 하여 리스트의 구조를 원형으로 만든 연결 리스트
 - 단순 연결 리스트의 마지막 노드의 링크 필드에 첫 번째 노드의 주소를 저장하여 구성
 - 링크를 따라 계속 순회하면 이전 노드에 접근 가능

1 원형 연결 리스트 특징

1 원형 연결 리스트의 개념



※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

1 | 원형 연결 리스트 특징

2 선형 기차 놀이와 원형 기차놀이 비교

▶ 기차 놀이에 대한 노드 표현

[선형 기차 놀이]



기차 놀이에 대한
노드 표현



※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

1 | 원형 연결 리스트 특징

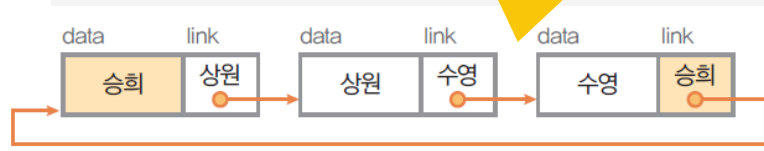
2 선형 기차 놀이와 원형 기차놀이 비교

▶ 기차 놀이에 대한 노드 표현



[원형 기차 놀이]

기차 놀이에 대한
노드 표현



※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

2 | 원형 연결 리스트 삽입과 삭제

1 원형 연결 리스트의 알고리즘

- ▶ 단순 연결 리스트에서 마지막 노드가 리스트의 첫 번째 노드를 가리키게 하여 리스트의 구조를 원형으로 만든 연결 리스트
- ▶ 리스트에 첫 번째 노드로 삽입하는 알고리즘

2 원형 연결 리스트 삽입과 삭제

1 원형 연결 리스트의 알고리즘

알고리즘 4-6 원형 연결 리스트의 첫 번째 노드 삽입

```
insertFirstNode(CL, x)
    new ← getNode();
    new.data ← x;
    if (CL = NULL) then {
1      1-a CL ← new;
        1-b new.link ← new;
    }

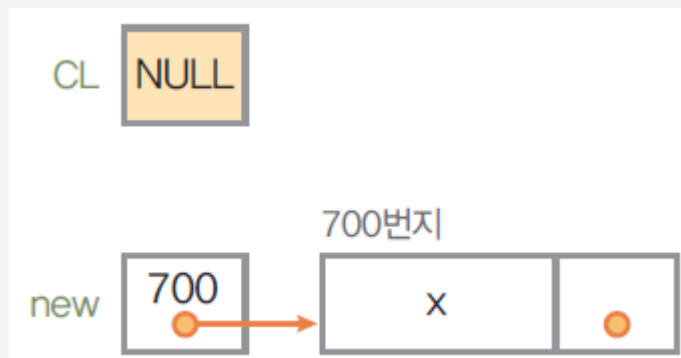
    else {
        2-a temp ← CL;
        while (temp.link ≠ CL) do
2      2-b temp ← temp.link;
        2-c new.link ← temp.link;
        2-d temp.link ← new;
        2-e CL ← new;
    }
end insertFirstNode()
```

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

- 1 공백 리스트인 경우
삽입하는 노드 new는 리스트의 첫 번째 노드이자
마지막 노드가 되어야 함

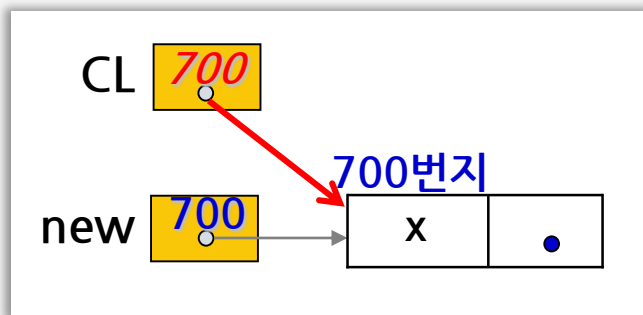


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

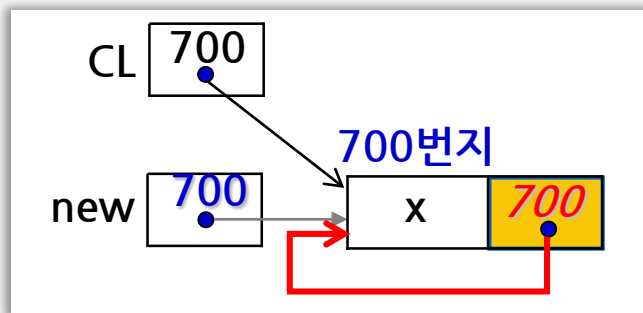
- ①-a $CL \leftarrow new;$
포인터 CL이 노드 new를 가리키게 함



2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

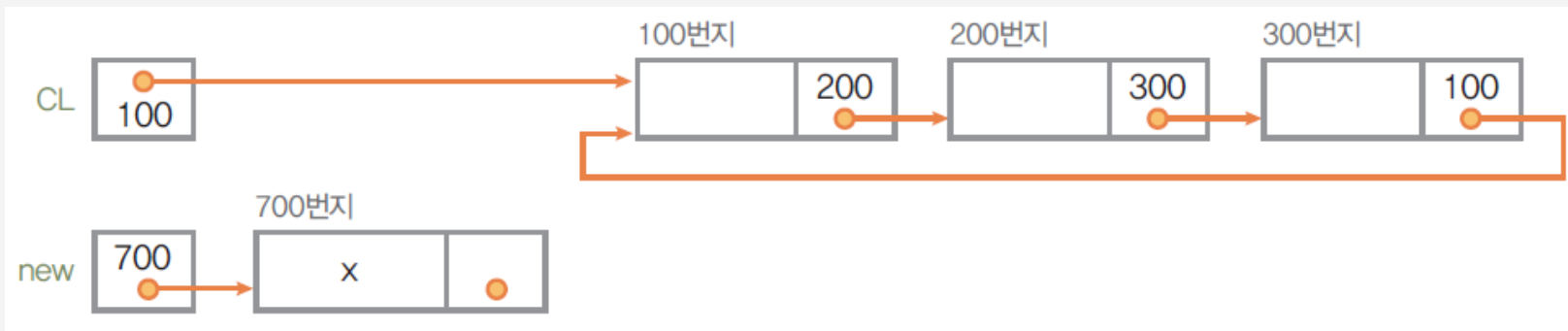
- ①-b `new.link ← new;`
노드 new가 자기자신을 가리키게 함으로써 노드 new를 첫 번째 노드이자 마지막 노드가 되도록 지정



2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

2 공백 리스트가 아닌 경우



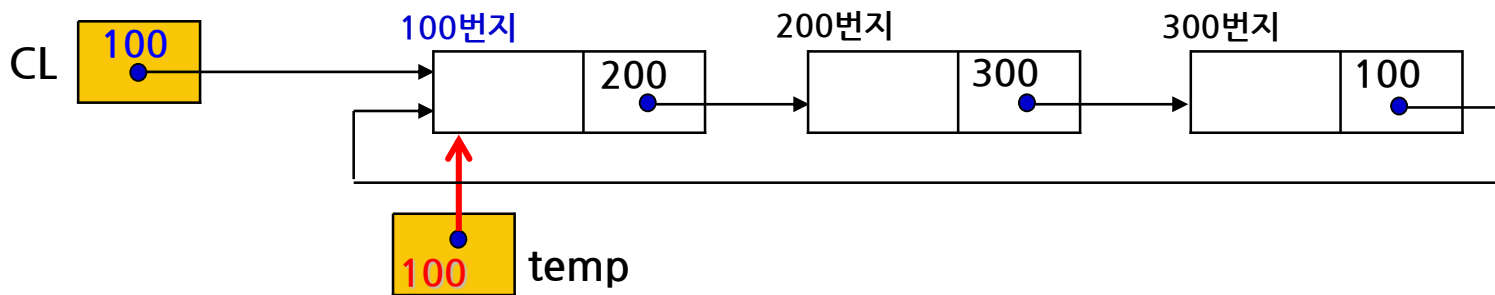
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

2-a `temp ← CL;`

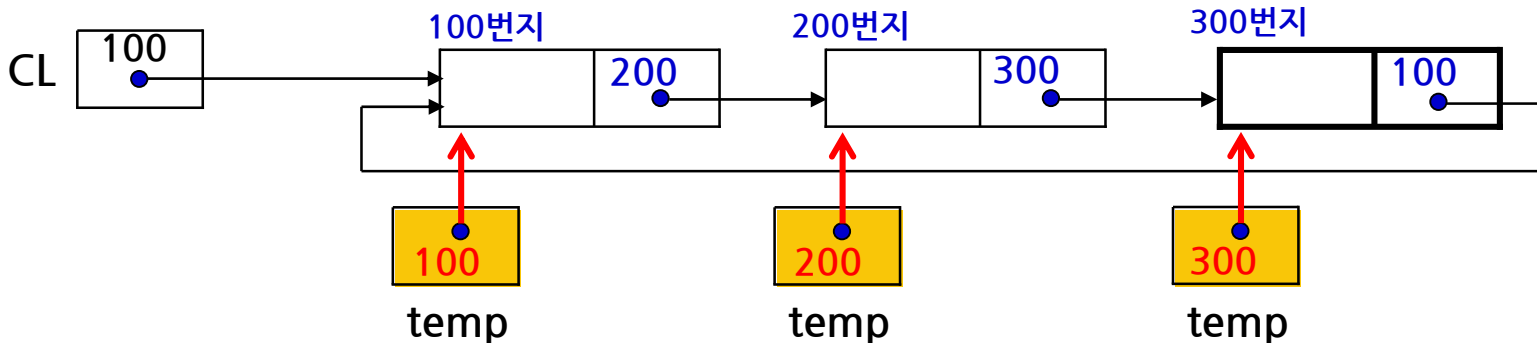
리스트가 공백리스트가 아닌 경우에는 첫 번째
노드의 주소를 임시 순회 포인터 temp에 저장하여
노드 순회의 시작점을 지정



2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

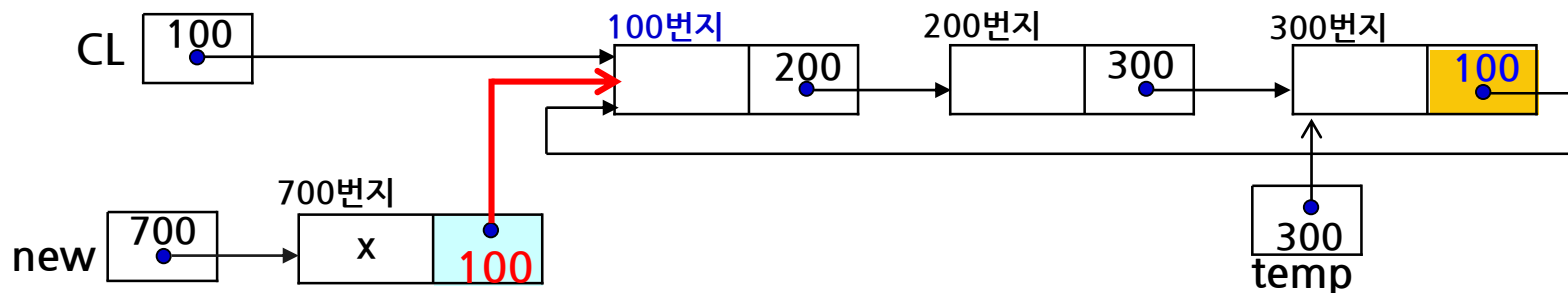
②-b `temp ← temp.link;`
while 문을 수행해 순회 포인터 temp를 링크를 따라
마지막 노드까지 이동



2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

② -c `new.link ← temp.link;`
리스트의 마지막 노드의 링크 값을 노드 new의
링크에 저장하여 노드 new는 노드 temp의 다음
노드인 첫 번째 노드와 연결

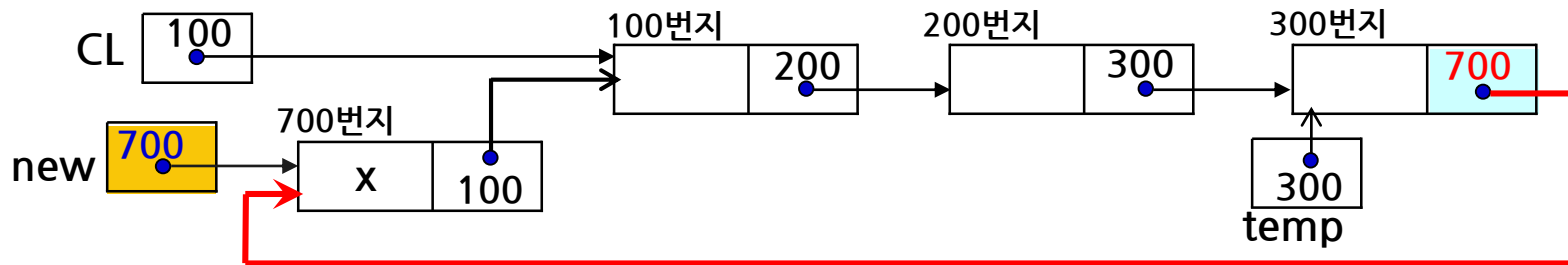


2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

②-d temp.link ← new;

포인터 new의 값을 포인터 temp가 가리키고 있는
마지막 노드의 링크에 저장하여, 리스트의 마지막
노드가 노드 new를 가리키게 한

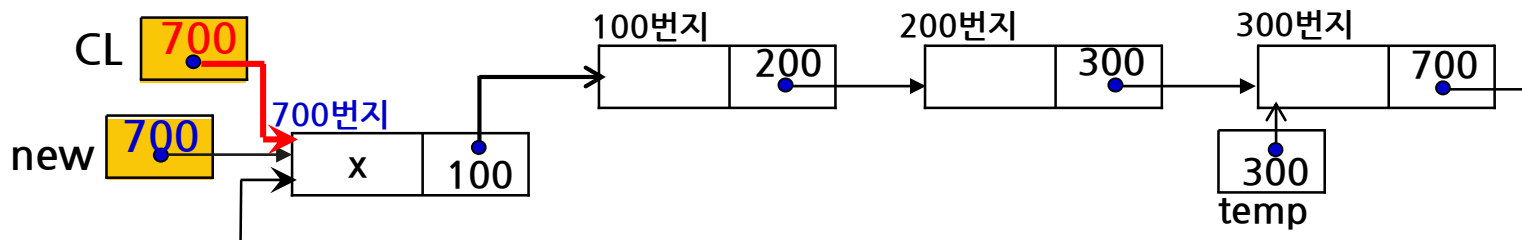


2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

②-e $CL \leftarrow new;$

노드 new 의 값을 리스트 포인터 CL 에 저장하여
노드 new 가 리스트의 첫 번째 노드가 되도록 지정



2 | 원형 연결 리스트 삽입과 삭제

2 원형 연결 리스트 첫 번째 노드로 삽입하는 과정

▶ 최종결과



[원형 연결 리스트 CL이 공백이 아닌 경우에 첫 번째 노드로 삽입한 최종 결과]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

알고리즘 4-7 원형 연결 리스트의 중간 노드 삽입

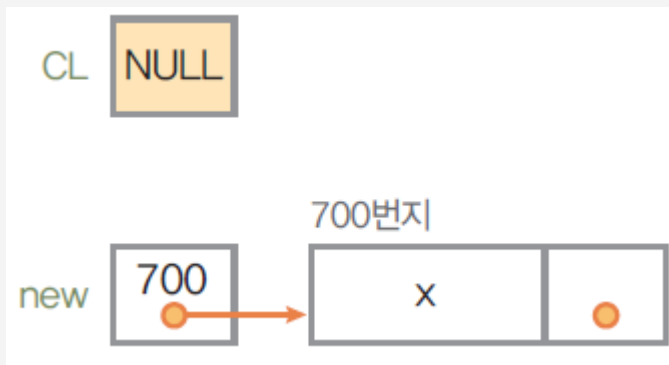
```
insertMiddleNode(CL, pre, x)
    new ← getNode();
    new.data ← x;
    {
        ① if (CL = NULL) then {
            CL ← new;
            new.link ← new;
        }
        {
            ② else {
                ②-a new.link ← pre.link;
                ②-b pre.link ← new;
            }
        }
    end insertMiddleNode()
```

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

① 공백 리스트인 경우

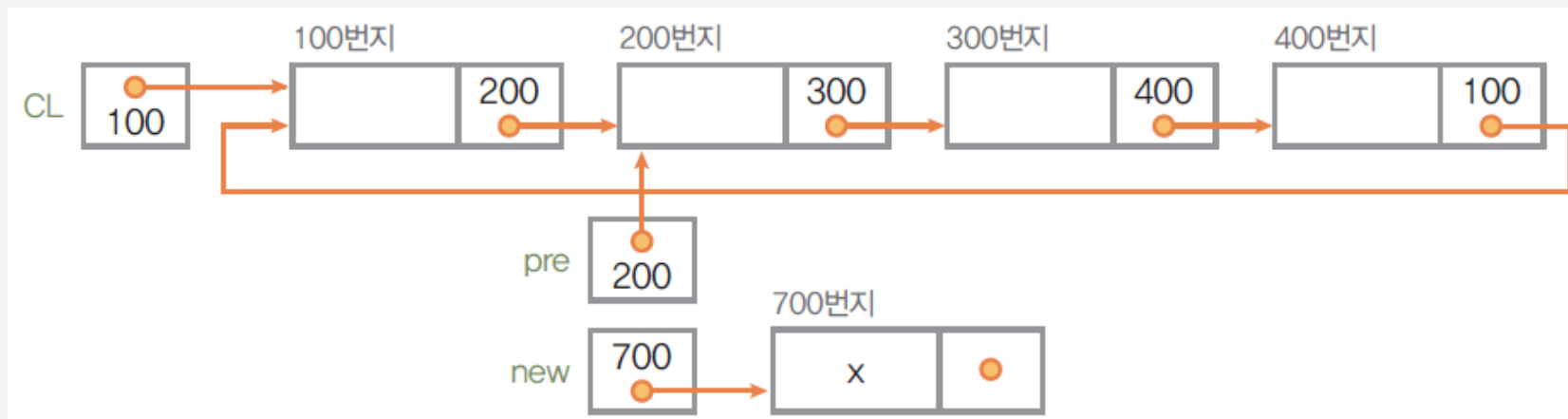


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

2 공백 리스트가 아닌 경우

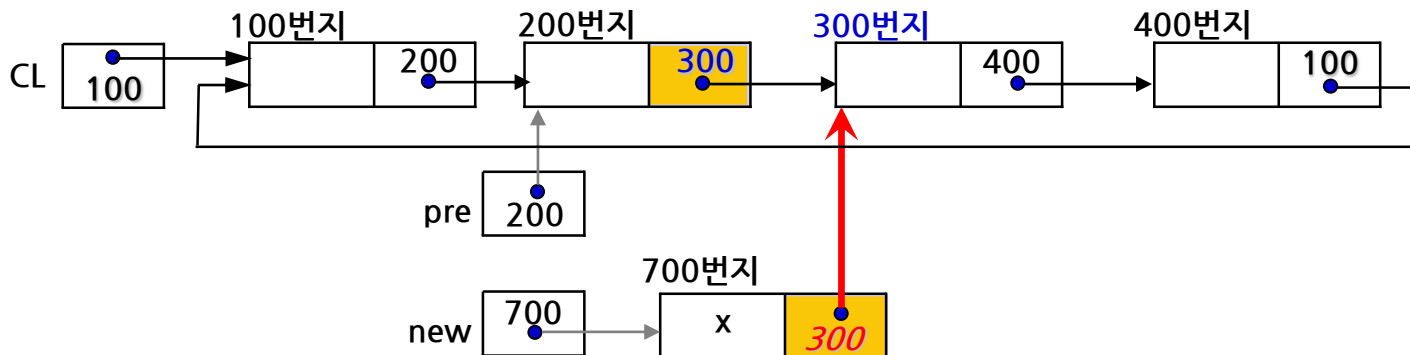


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

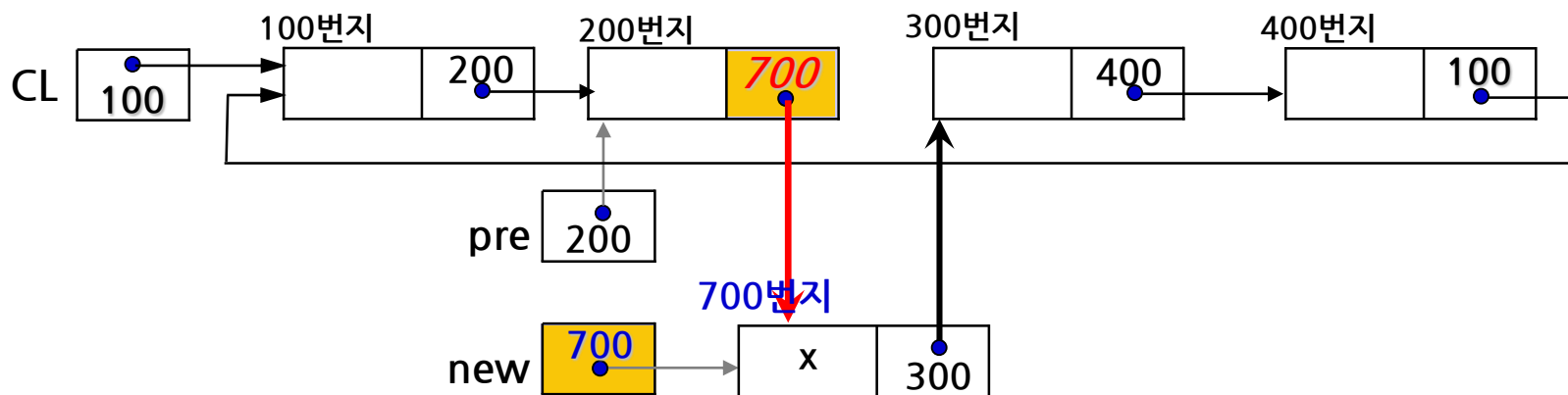
- ②-a $\text{new.link} \leftarrow \text{pre.link};$
노드 pre의 다음 노드로 new를 삽입하기 위해서,
먼저 노드 pre의 다음 노드(pre.link)를 new의 다음
노드(new.link)로 연결



2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

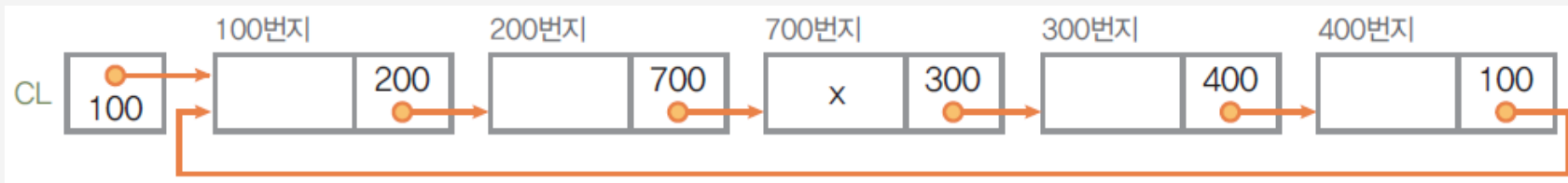
- ②-b $pre.link \leftarrow new$;
노드 new의 값(삽입할 노드의 주소)을 노드 pre의
링크에 저장하여, 노드 pre가 노드 new를 가리키도록 함



2 | 원형 연결 리스트 삽입과 삭제

3 리스트 중간에 노드를 삽입하는 알고리즘

▶ 최종결과



[원형 연결 리스트 CL이 공백이 아닌 경우에 중간 노드로 삽입한 최종 결과]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

4 노드 삭제 알고리즘

알고리즘 4-8 원형 연결 리스트의 노드 삭제

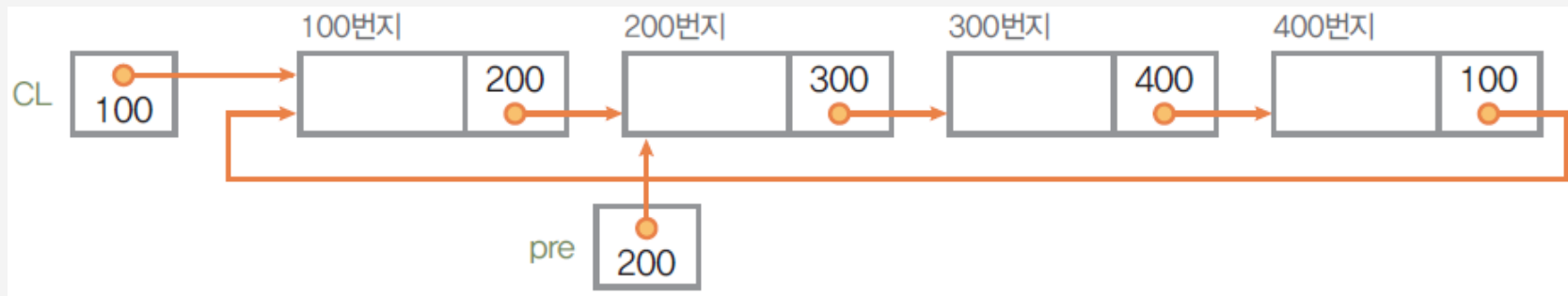
```
deleteNode(CL, pre)
  if (CL = NULL) then error;
  else {
    ❶ old ← pre.link;
    ❷ pre.link ← old.link;
    ❸ { if (old = CL) then
        ❸-a CL ← old.link;
    ❹ returnNode(old);
  }
end deleteNode()
```

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

▶ 초기상태

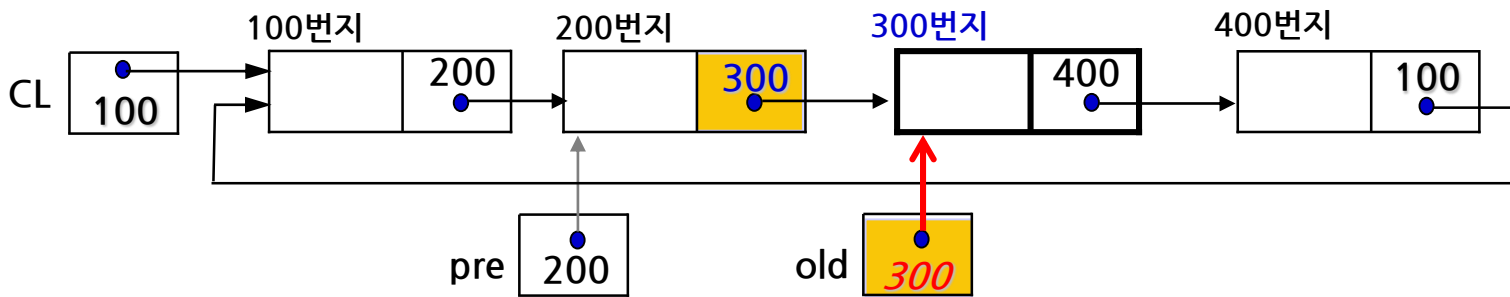


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

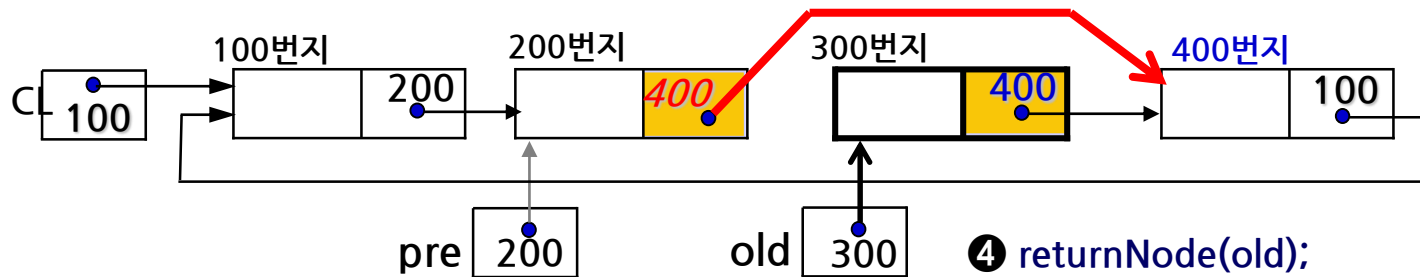
- ① $old \leftarrow pre.link;$
노드 pre 의 다음노드($pre.link$)를 삭제할 노드 old 로 지정



2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

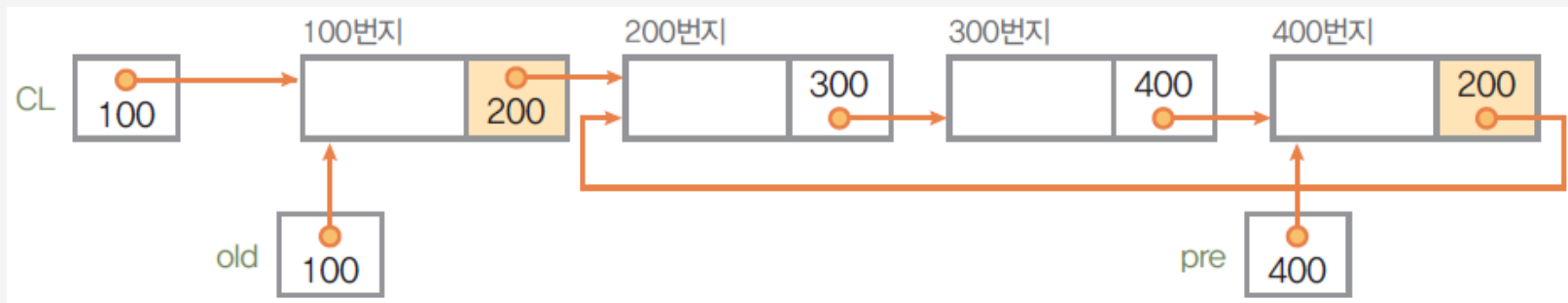
- ② $\text{pre.link} \leftarrow \text{old.link};$
old의 다음노드(old.link)를 노드 old의 이전노드
pre 노드와 서로 연결



2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

- ③ if (old = CL) then
삭제할 노드 old가 원형 연결 리스트의 첫번째 노드인 경우 >>



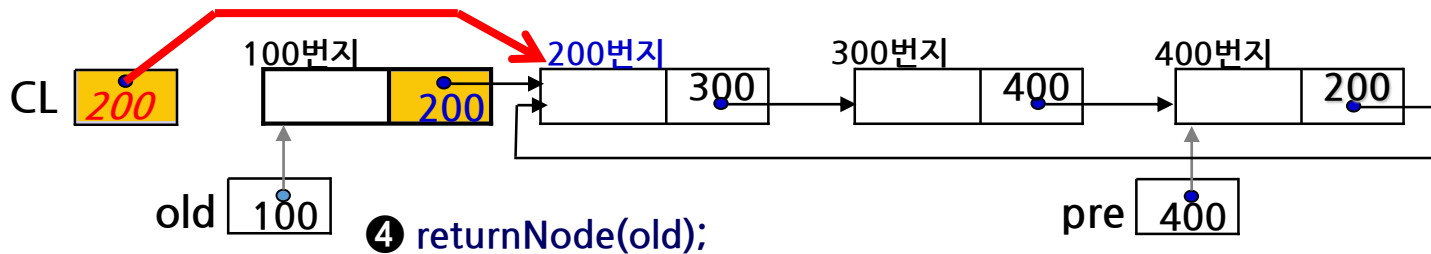
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

3-a $CL \leftarrow old.link;$

첫 번째 노드를 삭제하는 경우에는 노드 old의 링크 값을 리스트 포인터 CL에 저장하여 두 번째 노드가 리스트의 첫 번째 노드가 되도록 조정



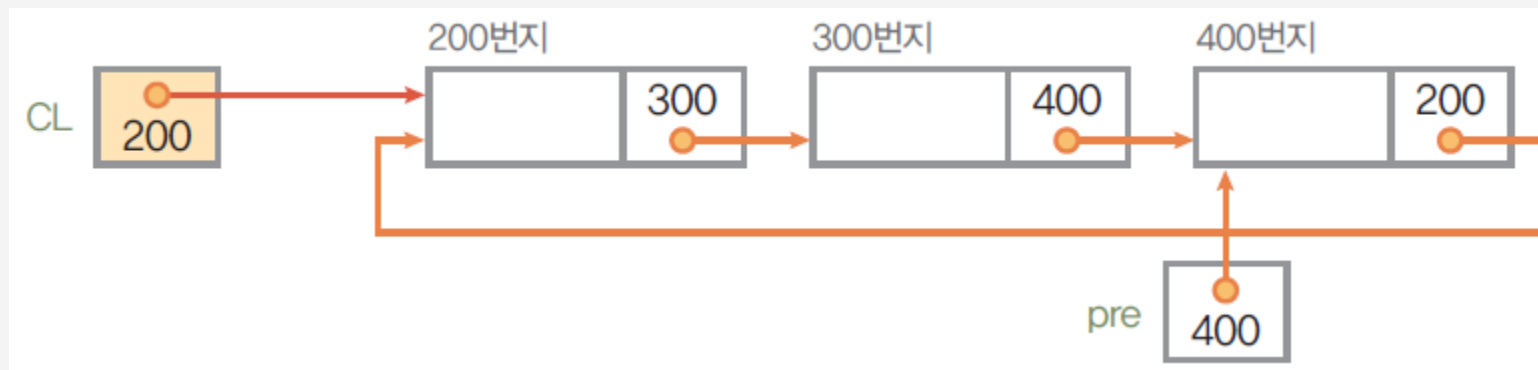
4 $returnNode(old);$

삭제한 노드 old의 메모리를 반환함

2 | 원형 연결 리스트 삽입과 삭제

5 원형 연결 리스트에서 노드를 삭제하는 과정

▶ 최종결과



[원형 연결 리스트에서 노드를 삭제한 최종 결과]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 이중 연결 리스트 특징

1 이중 연결 리스트의 개념

- ▶ 양쪽 방향으로 순회할 수 있도록 노드를 연결한 리스트
- ▶ 이중 연결 리스트의 노드 구조와 구조체 정의
 - llink(left link) 필드
: 왼쪽노드와 연결하는 포인터
 - rlink(right link) 필드
: 오른쪽 노드와 연결하는 포인터

3 | 이중 연결 리스트 특징

1 이중 연결 리스트의 개념

[이중 연결 리스트의 노드 구조와 구조체 정의]



(a) 노드 구조

```
typedef struct Dnode {  
    struct Dnode *llink;  
    char data[5];  
    struct Dnode *rlink;  
}
```

(b) 구조체 정의

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 이중 연결 리스트 특징

2 단방향과 양방향 기차놀이



(a) 단방향 기차놀이



(b) 양방향 기차놀이

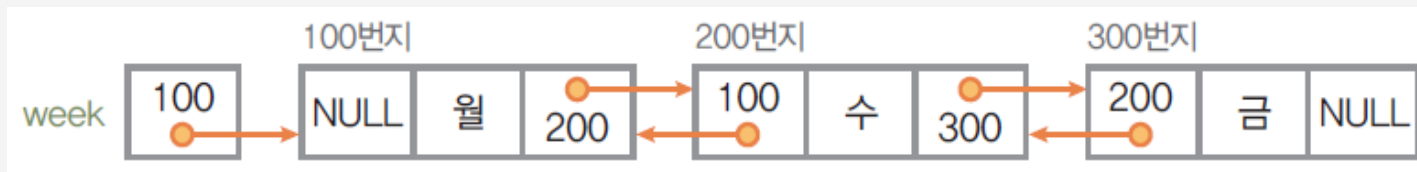
(b)의 양방향 기차를
이중 연결 리스트라고 생각하면
아이들은 노드 왼손의 이름표는 llink,
오른손의 이름표는 rlink

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

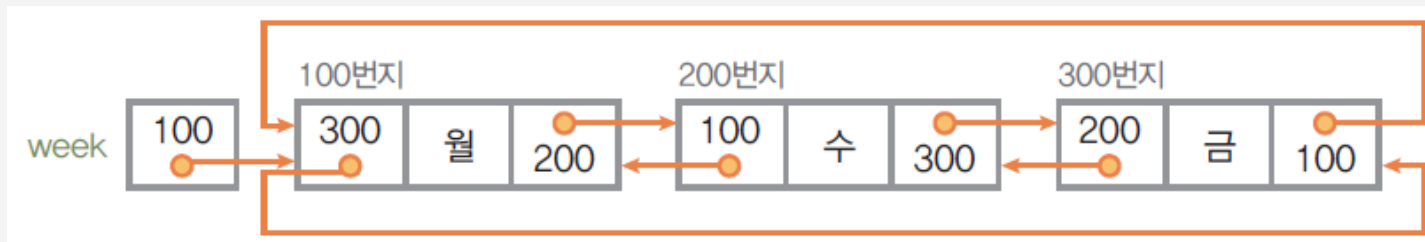
3 | 이중 연결 리스트 특징

3 원형 이중 연결 리스트

▶ 리스트 week=(월, 수, 금)의 이중 연결 리스트 구성



▶ 이중 연결 리스트를 원형으로 구성



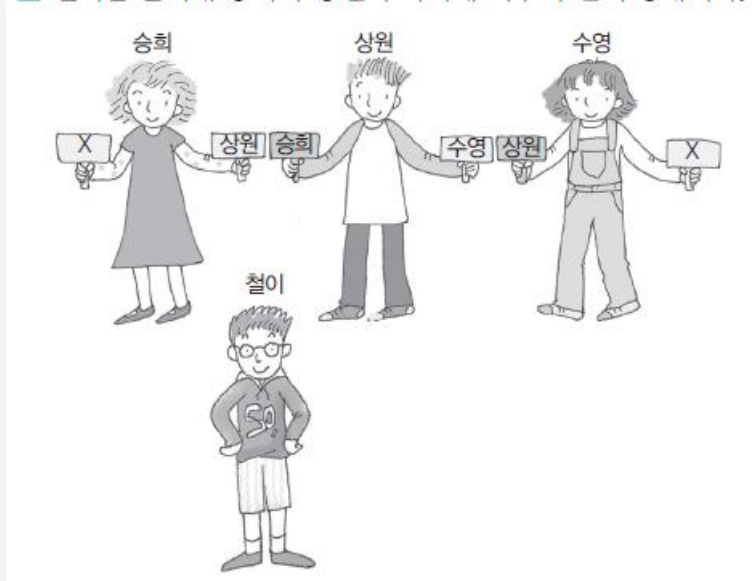
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

❶ 철이를 불러내 승희와 상원이 사이에 끼우기 전의 상태이다.

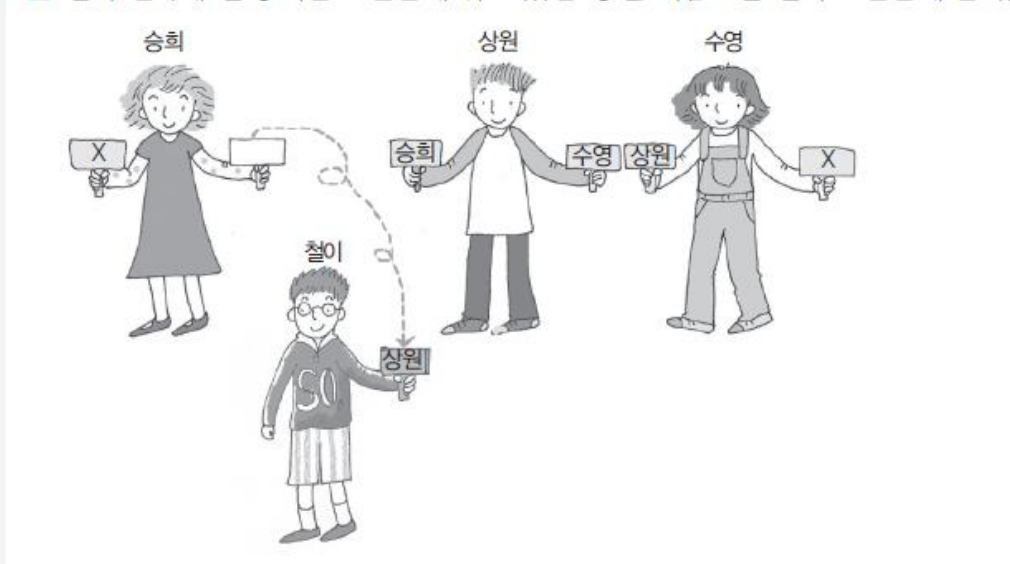


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

2 철이 왼쪽에 설 승희는 오른손에 쥐고 있던 '상원' 이름표를 철이 오른손에 준다.

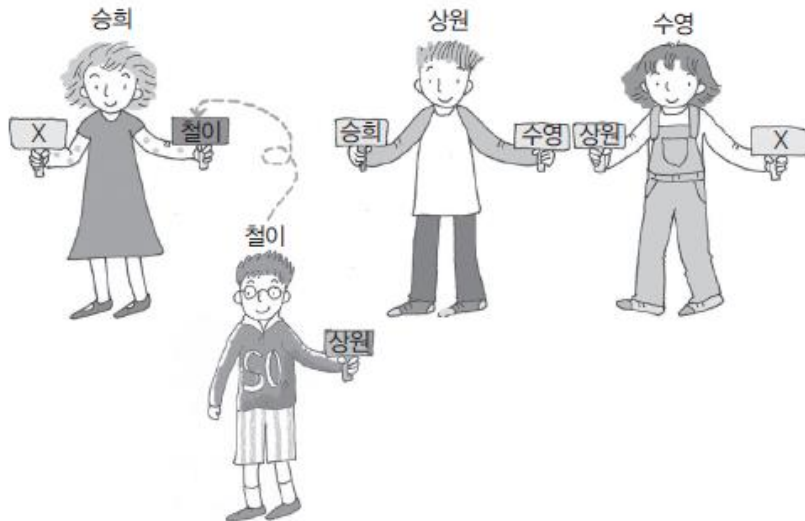


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

3 대신 철이는 '철이' 이름표를 승희에게 준다.

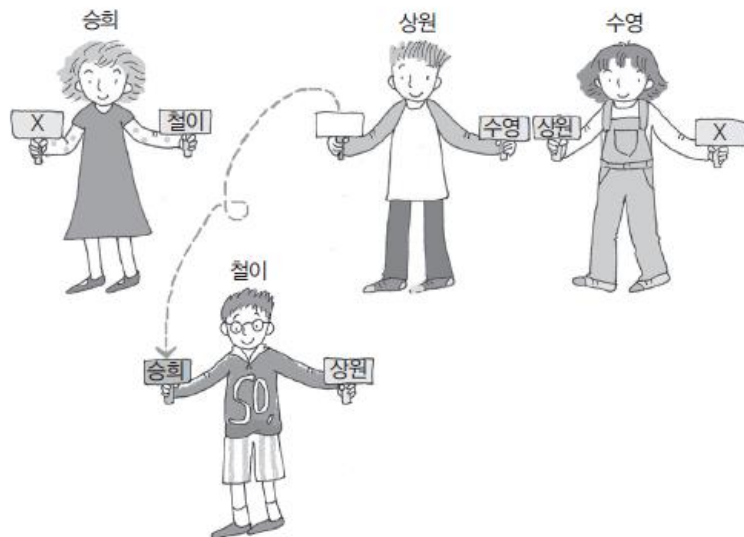


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

4 철이 오른쪽에 설 상원이는 왼손에 쥐고 있던 '승희' 이름표를 철이 왼손에 준다.

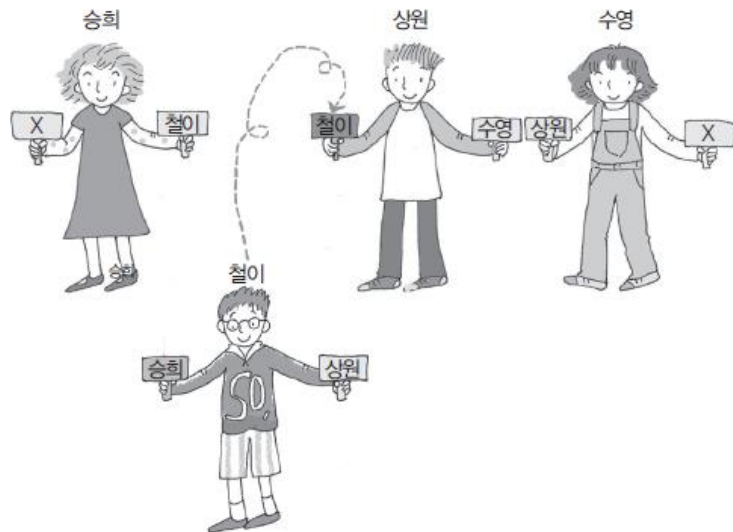


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

5 대신 철이는 '철이' 이름표를 상원에게 준다.

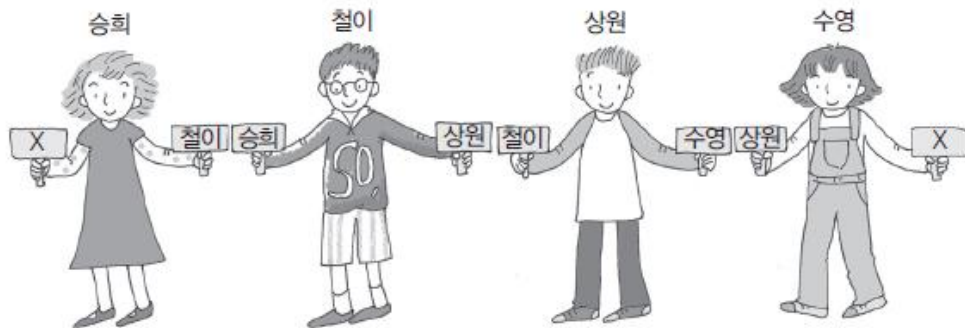


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

1 이중 연결 리스트에 노드를 삽입하는 예

6 이름표대로 연결하면 다음과 같은 양방향 기차가 완성된다.



※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

2 이중 연결 리스트에서 노드를 삽입하는 방법

- 1 삽입할 노드를 준비한다.
- 2 새 노드의 데이터 필드에 값을 저장한다.
- 3 새 노드 왼쪽 노드의 오른쪽 링크 필드(rlink)에 있던 값을 새 노드의 오른쪽 링크 필드(rlink)에 저장한다.
- 4 왼쪽 노드의 오른쪽 링크 필드(rlink)에 새 노드의 주소를 저장한다.
- 5 새 노드 오른쪽 노드의 왼쪽 링크 필드(llink)에 있던 값을 새 노드의 왼쪽 링크 필드(llink)에 저장한다.
- 6 오른쪽 노드의 왼쪽 링크 필드(llink)에 새 노드의 주소를 저장한다.
- 7 노드를 순서대로 연결한다.

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

3 이중 연결 리스트에 노드를 삽입하는 알고리즘

알고리즘 4-9 이중 연결 리스트의 노드 삽입

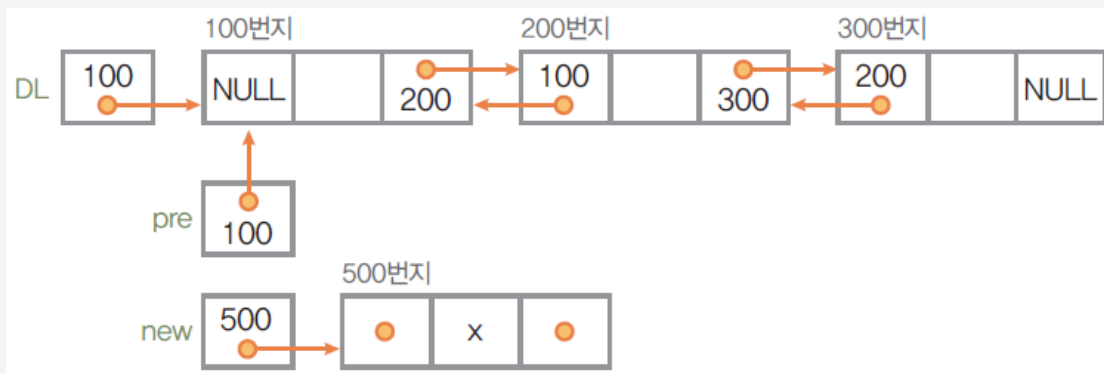
```
insertNode(DL, pre, x)
    new ← getNode();
    new.data ← x;
    ① new.rlink ← pre.rlink;
    ② pre.rlink ← new;
    ③ new.llink ← pre;
    ④ new.rlink.llink ← new;
end insertNode()
```

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

- ▶ 데이터 필드값이 x인 노드 new를 준비, 포인터 pre가 가리키는 노드의 다음 노드로 삽입하려고 한다고 가정
- ▶ 초기상태

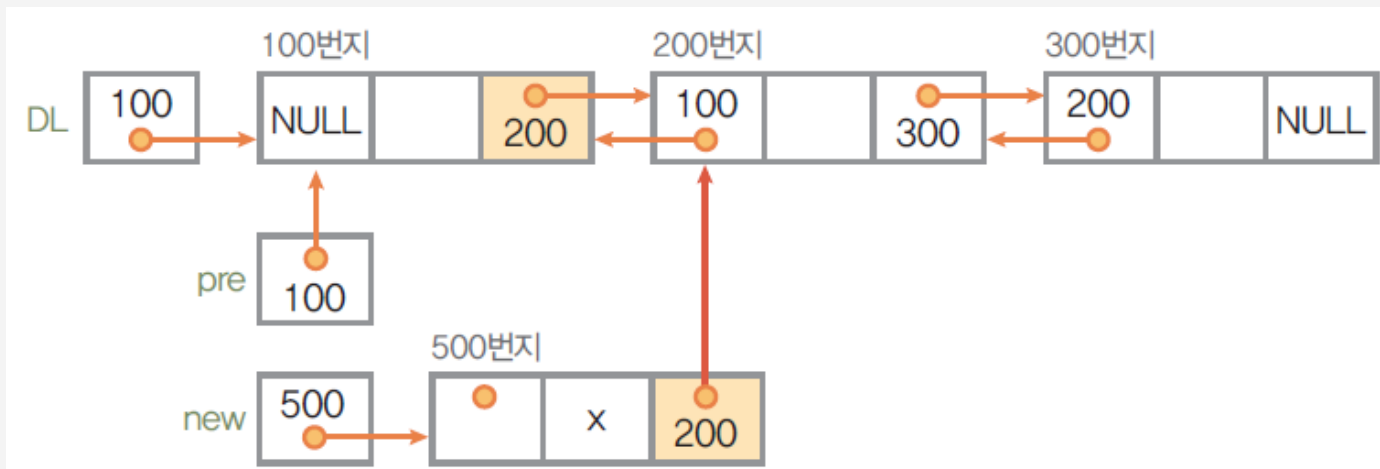


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

❶ `new.rlink ← pre.rlink;`

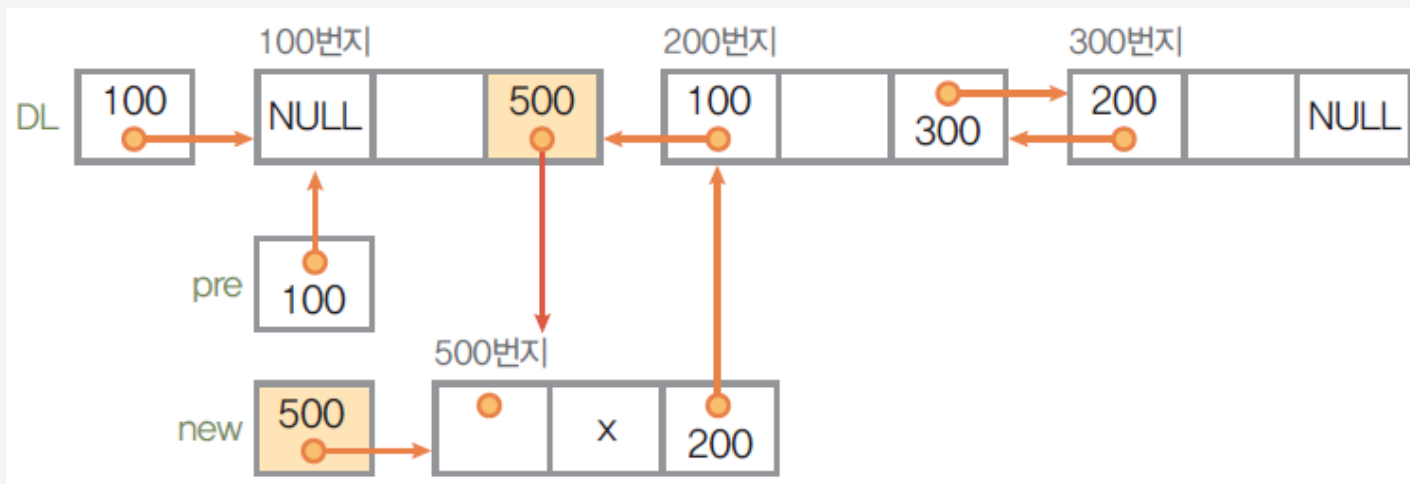


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

2 $\text{pre.rlink} \leftarrow \text{new};$

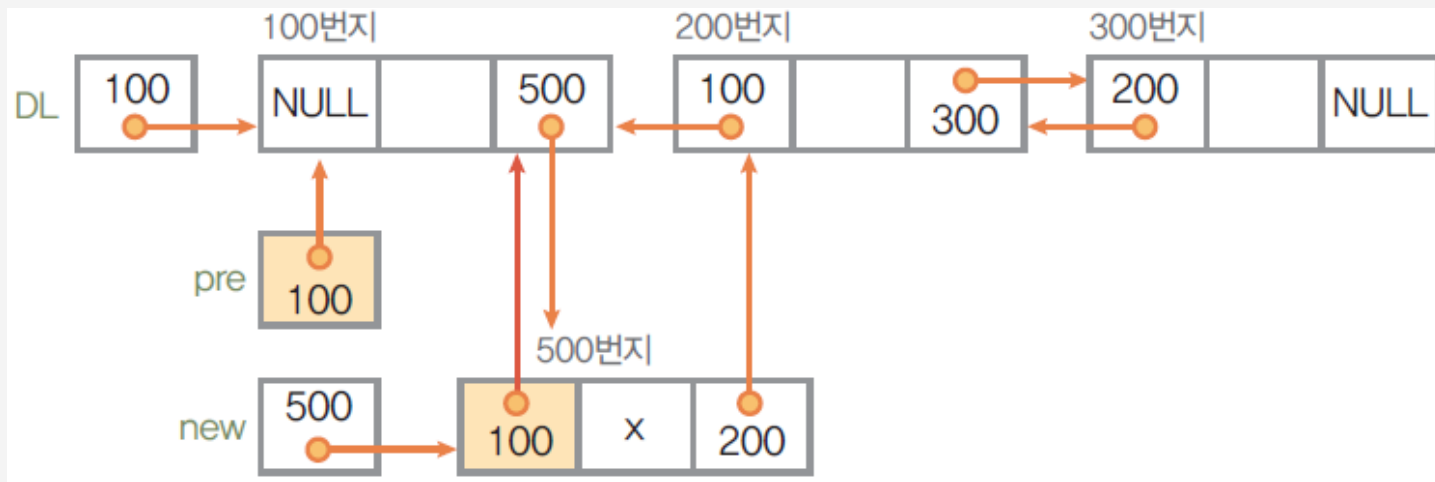


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

3 new.link ← pre;

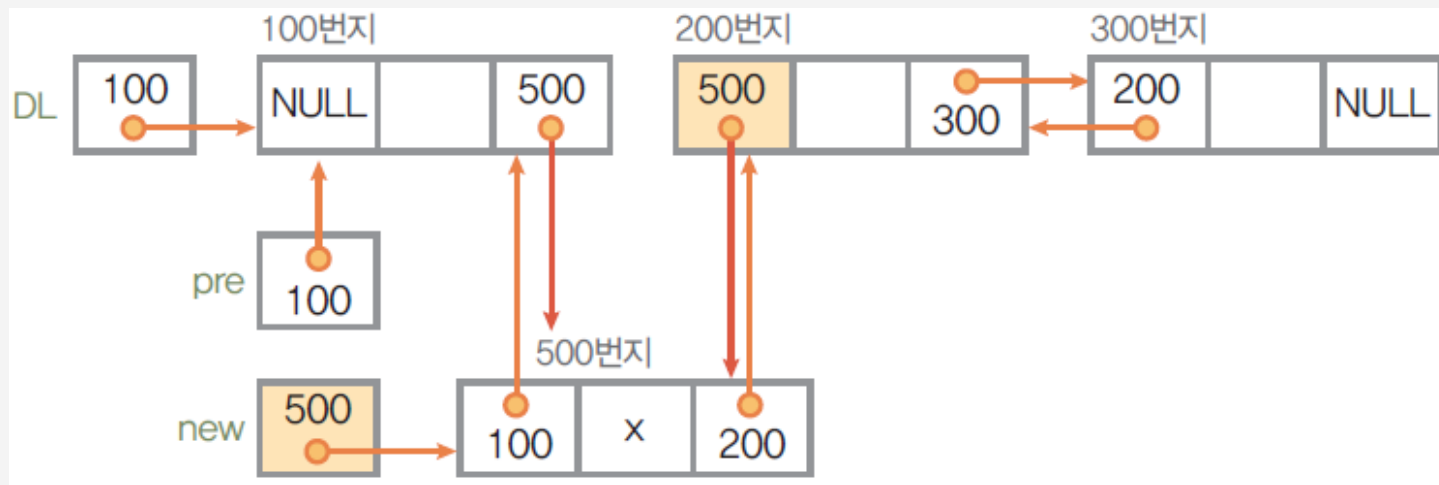


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

④ `new.rlink.llink ← new;`

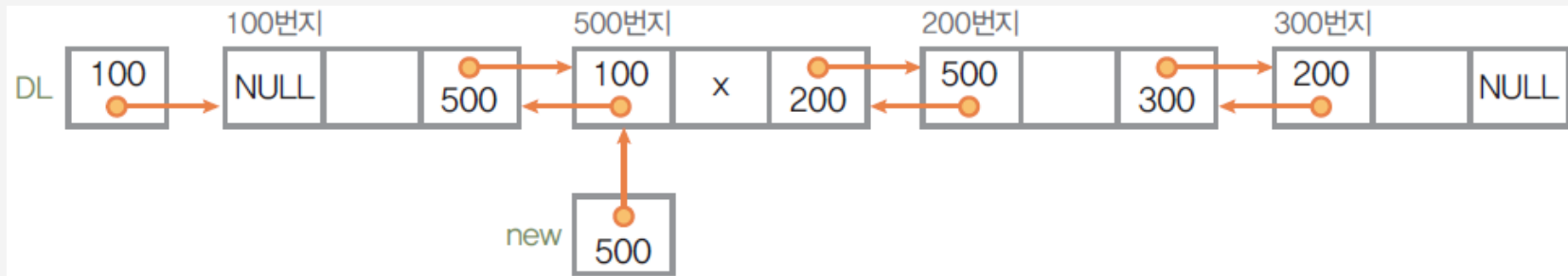


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

4 이중 연결 리스트에 노드를 삽입하는 과정

▶ 최종결과



[이중 연결 리스트 DL에 new 노드가 삽입된 최종 결과]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

5 이중 연결 리스트에서 노드를 삭제하는 방법

- 1 삭제할 노드의 오른쪽 노드와 왼쪽 노드를 찾는다.
- 2 삭제할 노드의 오른쪽 노드의 주소(old.rlink)를 삭제할 노드의 왼쪽 노드(old.llink)의 오른쪽 링크 필드(rlink)에 저장한다.
- 3 삭제할 노드의 왼쪽 노드(old.llink)의 주소를 삭제할 노드의 오른쪽 노드(old.rlink)의 왼쪽 링크 필드에 저장한다.
- 4 노드를 순서대로 연결한다.

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

6 이중 연결 리스트에 노드를 삭제하는 알고리즘

알고리즘 4-10 이중 연결 리스트의 노드 삭제

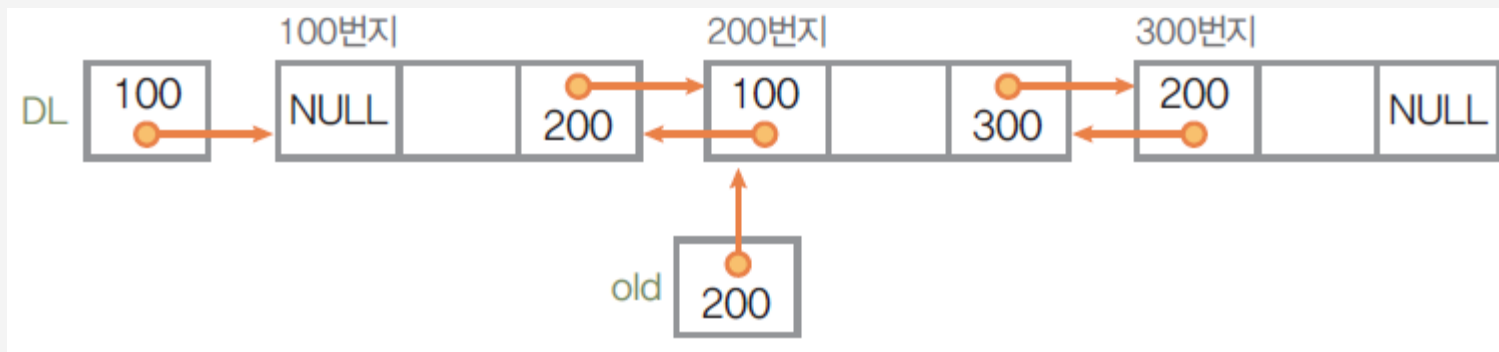
```
deleteNode(DL, old)
    ❶ old.llink.rlink ← old.rlink;
    ❷ old.rlink.llink ← old.llink;
    ❸ returnNode(old);
end deleteNode()
```

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

7 이중 연결 리스트에서 노드를 삭제하는 과정

▶ 초기상태

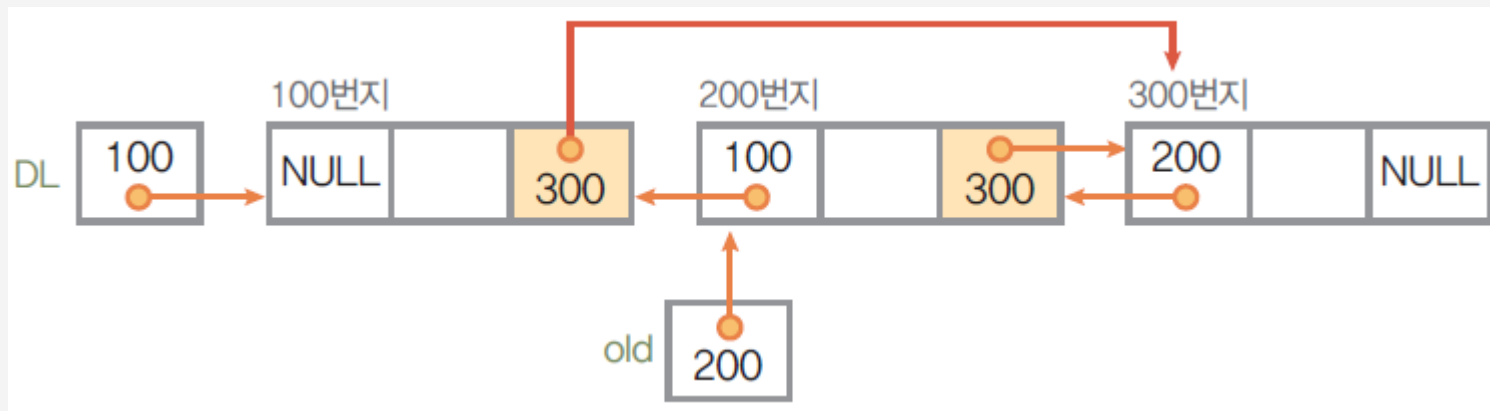


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

7 이중 연결 리스트에서 노드를 삭제하는 과정

❶ `old.llink.rlink ← old.rlink;`

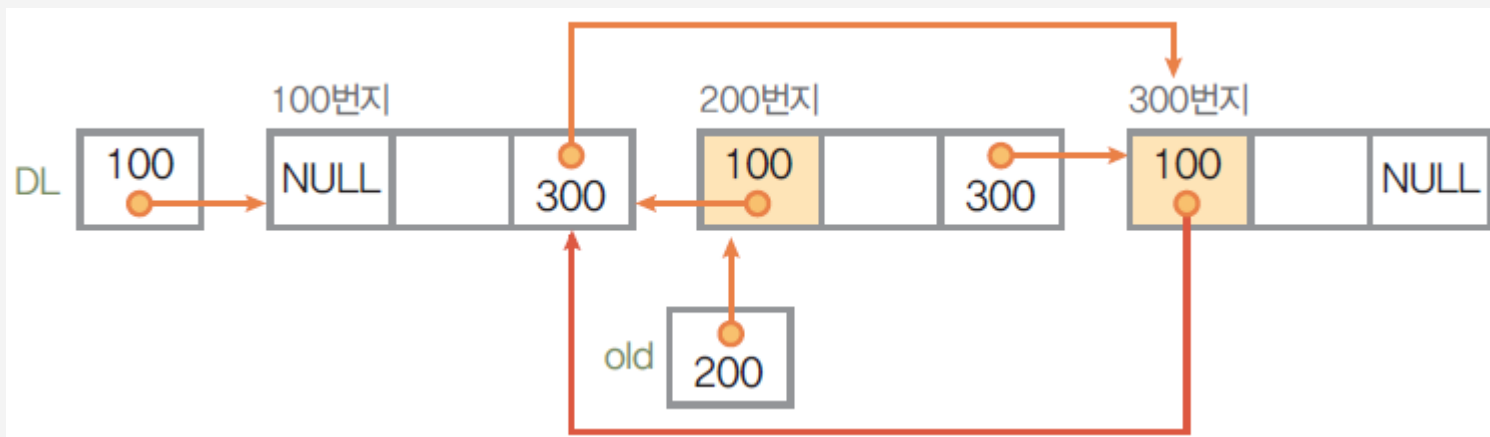


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 | 이중 연결 리스트 삽입과 삭제

7 이중 연결 리스트에서 노드를 삭제하는 과정

② `old.rlink.llink ← old.llink;`



③ `returnNode(old);`

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어



7 이중 연결 리스트에서 노드를 삭제하는 과정

▶ 최종결과



[이중 연결 리스트에서 old 노드를 삭제한 최종 결과]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어