1 단순 연결 리스트의 삽입방법과 알고리즘

1 단순 연결 리스트에서의 삽입 연산

▶ 승희와 상원 사이에 철이를 끼우려고 함

ⓞ 기차놀이에 철이를 끼우기 전 상태

① 앞사람에게 이름표 받아 오기

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

1 단순 연결 리스트에서의 삽입 연산

▶ 승희와 상원 사이에 철이를 끼우려고 함

② 앞사람에게 자기 이름표 주기

③ 이름표대로 연결하기

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

2 단순 연결 리스트에 삽입하는 방법

1. 삽입할 노드를 준비함
2. 새 노드의 데이터 필드에 값을 저장함
3. 새 노드의 링크값을 지정함
4. 리스트의 앞 노드에 새 노드를 연결함

2 단순 연결 리스트에 삽입하는 방법

- 단순 연결 리스트 week2=(월, 금, 일),
  '월'과 '금' 사이에 '수' 삽입 과정
  - 초기상태

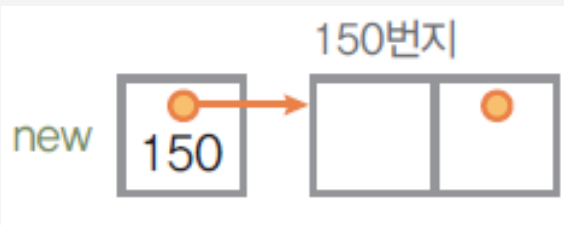
: 단순 연결 리스트 week2에 노드를 삽입하기 전인 초기 상태

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

2 단순 연결 리스트에 삽입하는 방법

- 단순 연결 리스트 week2=(월, 금, 일),
  '월'과 '금' 사이에 '수' 삽입 과정

  ❶ 삽입할 노드 준비
  : 공백 노드를 가져와 포인터 변수 new가 가리키게 함

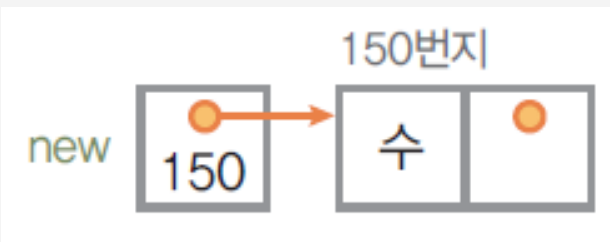


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 단순 연결 리스트에 삽입하는 방법

- 단순 연결 리스트 week2=(월, 금, 일),
  '월'과 '금' 사이에 '수' 삽입 과정

  ❷ 새 노드의 데이터 필드값 저장
  : new의 데이터 필드에 "수"를 저장

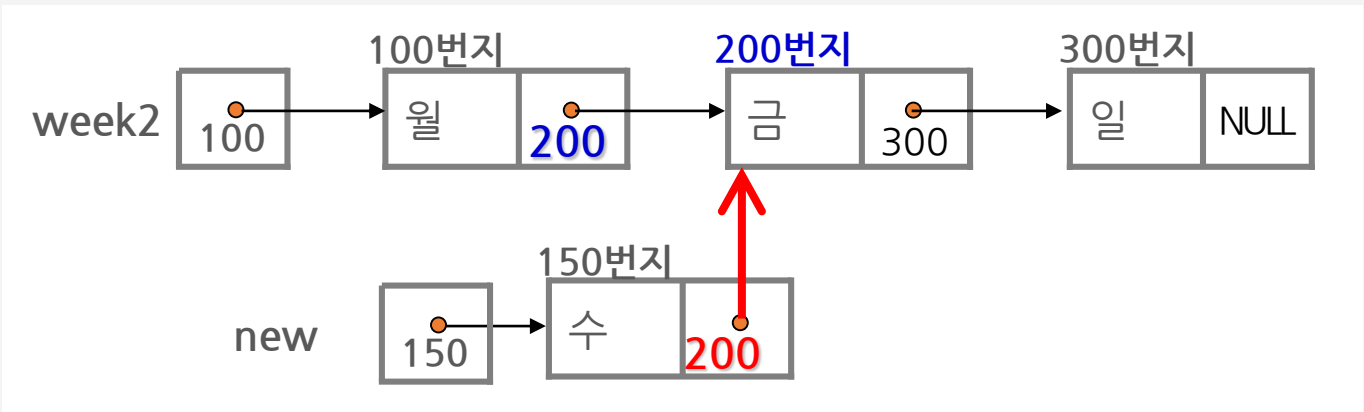


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 단순 연결 리스트에 삽입하는 방법

- 단순 연결 리스트 week2=(월, 금, 일),
  '월'과 '금' 사이에 '수' 삽입 과정

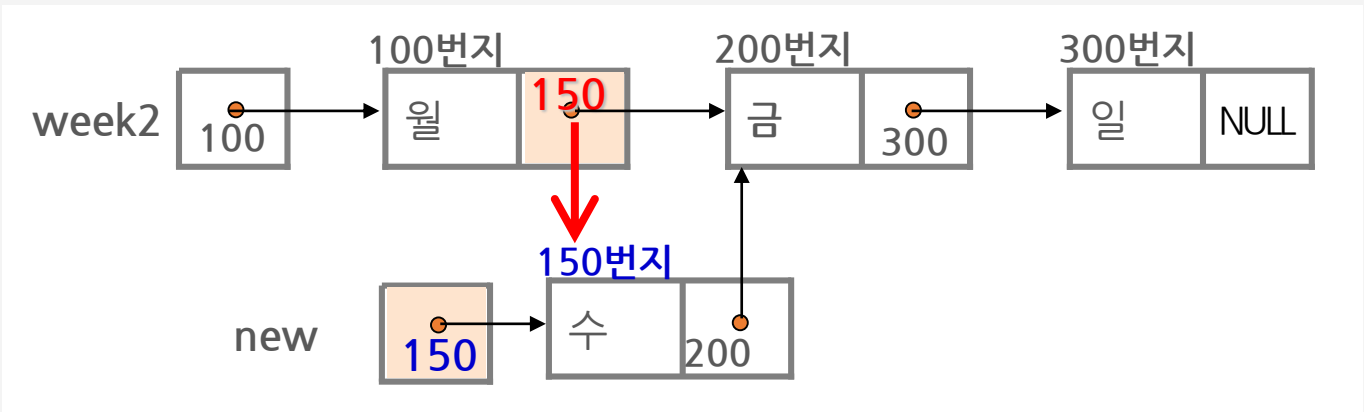
❸ 새 노드의 링크 필드값 지정
: new의 앞 노드, 즉 "월"노드의 링크 필드 값을 **new의 링크 필드**에 저장



2 단순 연결 리스트에 삽입하는 방법

- 단순 연결 리스트 week2=(월, 금, 일),
  '월'과 '금' 사이에 '수' 삽입 과정

❹ 리스트의 앞 노드에 새 노드 연결
: new의 값을 **"월"노드의 링크 필드**에 저장



3 단순 연결 리스트의 알고리즘 : 첫 번째 노드로 삽입하기

알고리즘 4-1 단순 연결 리스트의 첫 번째 노드 삽입

```
insertFirstNode(L, x)
    ❶ new ← getNode();
    ❷ new.data ← x;
    ❸ new.link ← L;
    ❹ L ← new;
end insertFirstNode()
```

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

3 단순 연결 리스트의 알고리즘 : 첫 번째 노드로 삽입하기

❶ new ← getNode();
삽입할 노드에 대한 메모리를 할당받아서 (getNode()), 포인터 new에 설정

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

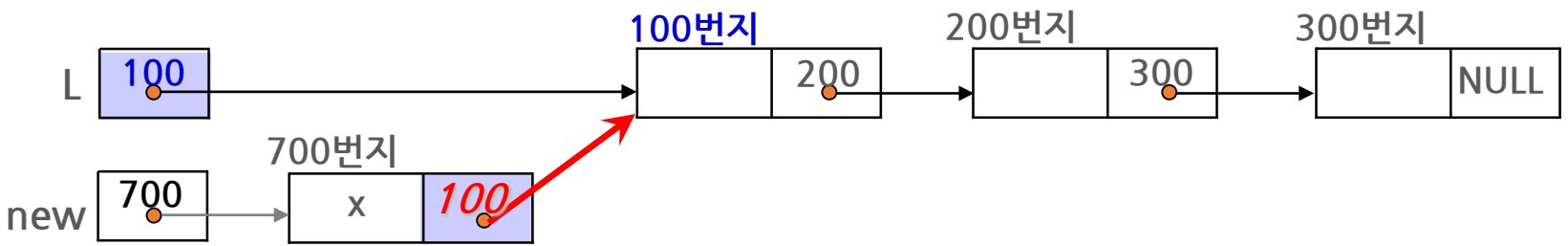
3 단순 연결 리스트의 알고리즘 : 첫 번째 노드로 삽입하기

❷ new.data ← x;
새 노드의 데이터 필드에 x를 저장

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

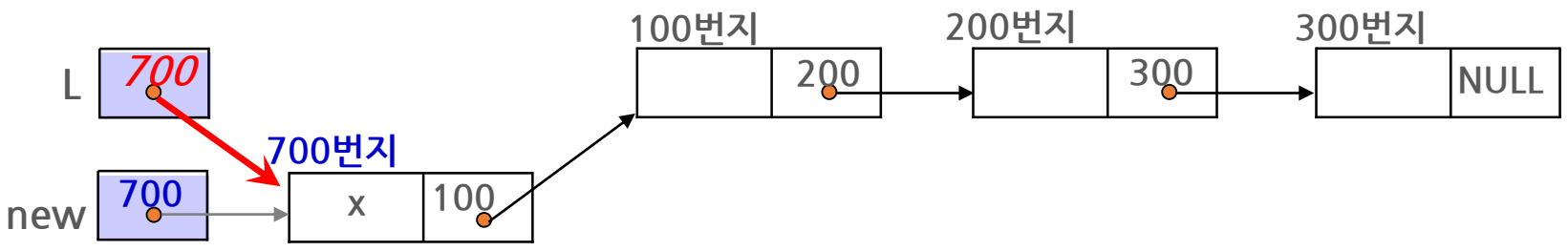
3 단순 연결 리스트의 알고리즘 : 첫 번째 노드로 삽입하기

❸ new.link ← L;
삽입할 노드를 연결하기 위해서 리스트의 첫 번째 노드 주소(L)를 삽입할 새 노드 **new의 링크 필드 (new.link)**에 저장하여, 새 노드 new가 리스트의 첫 번째 노드를 가리키게 함



3 단순 연결 리스트의 알고리즘 : 첫 번째 노드로 삽입하기

❹ L ← new;
리스트의 첫 번째 노드 주소를 저장하고 있는
포인터 L에, 새 노드의 주소 new를 저장하여,
포인터 L이 새 노드를 첫 번째 노드로 가리키도록
지정



4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

알고리즘 4-2 단순 연결 리스트의 중간 노드 삽입

```
insertMiddleNode(L, pre, x)
    ❶ new ← getNode();
    ❷ new.data ← x;
    ❸ if (L = NULL) then {
          ❸-ⓐ L ← new;
          ❸-ⓑ new.link ← NULL;
      }
    ❹ else {
          ❹-ⓐ new.link ← pre.link;
          ❹-ⓑ pre.link ← new;
}
end insertMiddleNode()
```

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

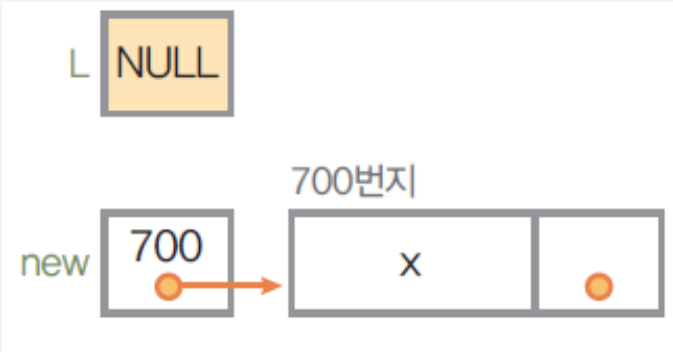
4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

- ❶ new← getNode();
  삽입할 노드에 대한 메모리를 할당 받아서
  (getNode()), 포인터 new에 설정

- ❷ new.data ← x;
  새 노드의 데이터 필드에 x를 저장

4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

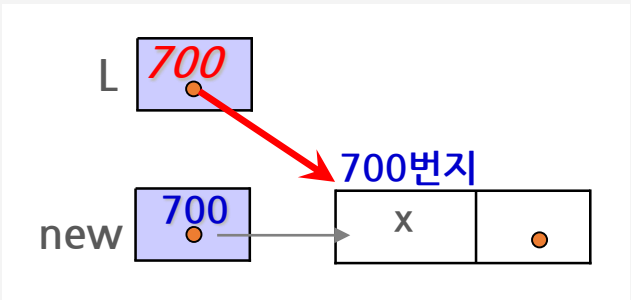
❸ 공백 리스트인 경우



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

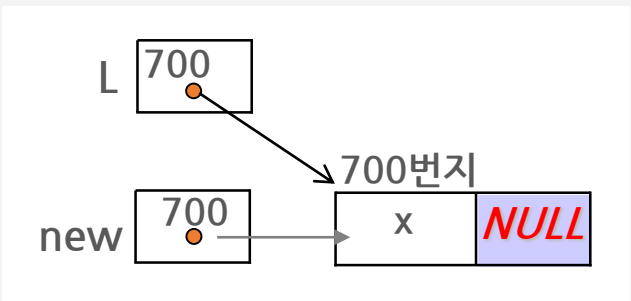
4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

- ❸- a L ← new;
  리스트 포인터 L에 새 노드 **new의 주소**를 저장하여,
  새 노드 new가 리스트의 첫 번째 노드가 되도록 함



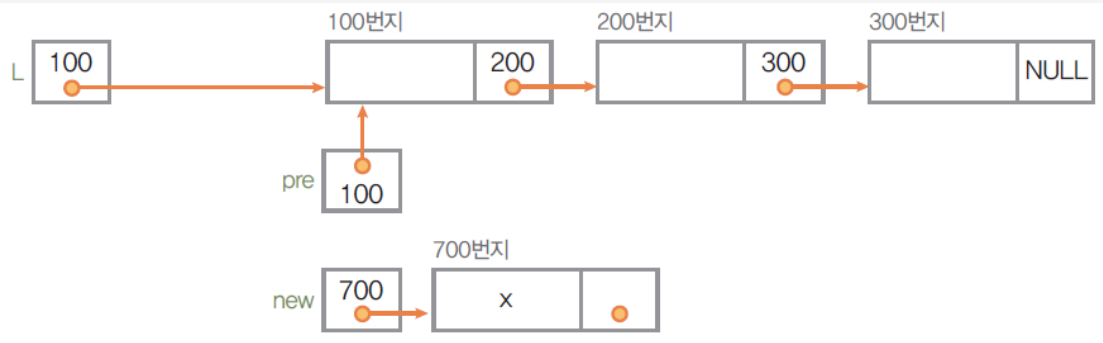
4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

- ❸- b new.link ← NULL;
  리스트의 마지막 노드인 **new의 링크 필드**에
  NULL을 저장해 마지막 노드임을 표시



4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

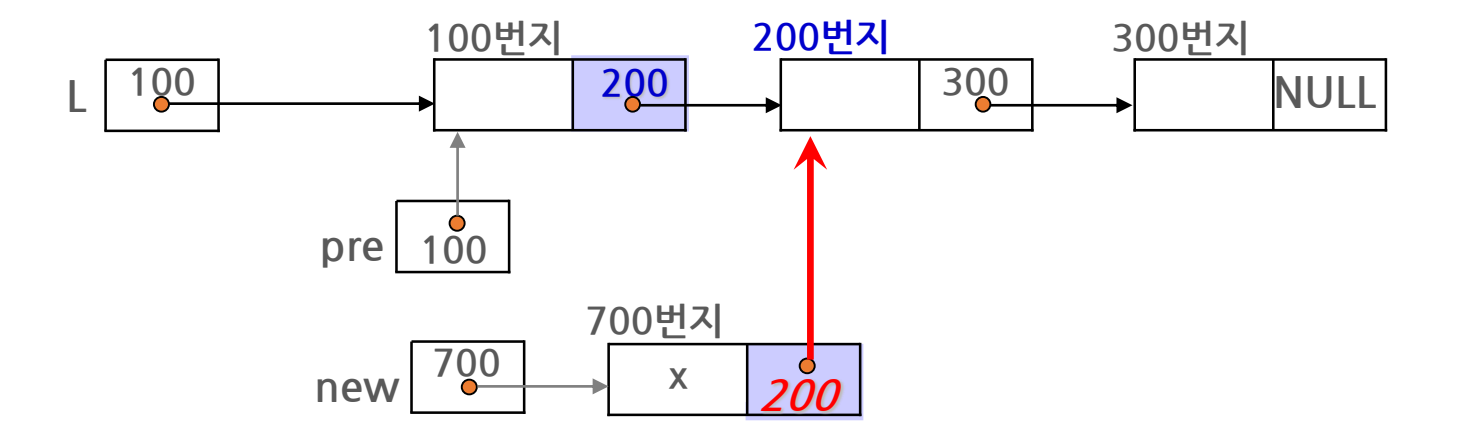
❹ 공백 리스트가 아닌 경우



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

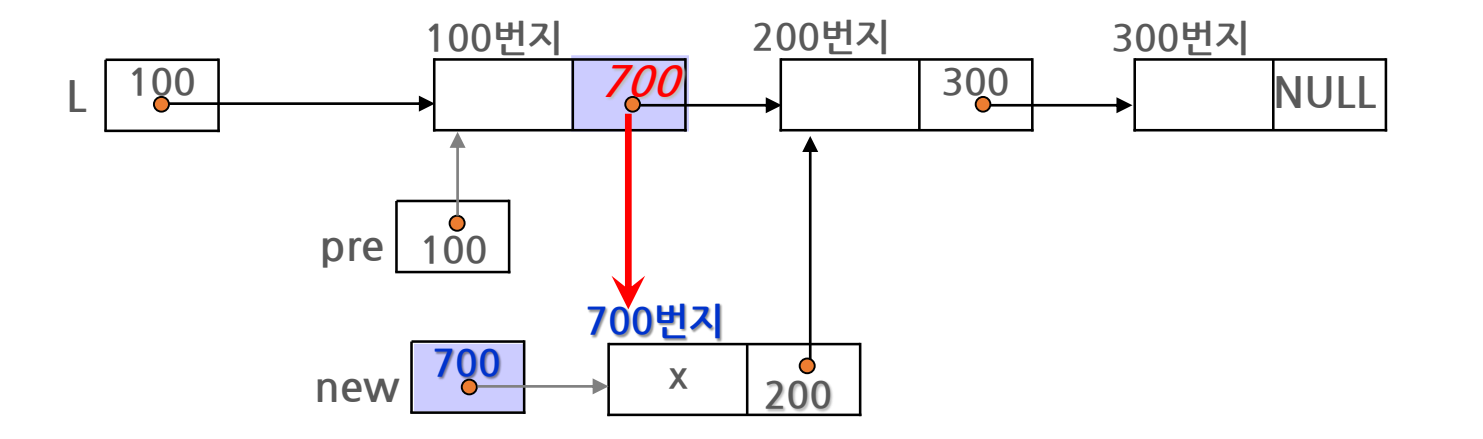
4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

❹- a new.link ← pre.link;
노드 pre의 링크 필드 값을 **노드 new의 링크 필드**에 저장하여, 새 노드 new가 노드 pre의 다음 노드를 가리키도록 함



4 단순 연결 리스트의 알고리즘 : 중간 노드로 삽입하기

❹- b pre.link ← new;
포인터 new의 값을 **노드 pre의 링크 필드**에 저장하여,
노드 pre가 새 노드 new를 다음 노드로 가리키도록 함



5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

알고리즘 4-3 단순 연결 리스트의 마지막 노드 삽입

```
insertLastNode(L, x)
    ❶ new ← getNode();
    ❷ new.data ← x;
    ❸ new.link ← NULL;
    ❹ if (L = NULL) then {
          ❹-ⓐ L ← new;
              return;
      }
    ❺ ❺-ⓐ temp ← L;
          while (temp.link ≠ NULL) do
              ❺-ⓑ temp ← temp.link;
      ❺-ⓒ temp.link ← new;
end insertLastNode()
```

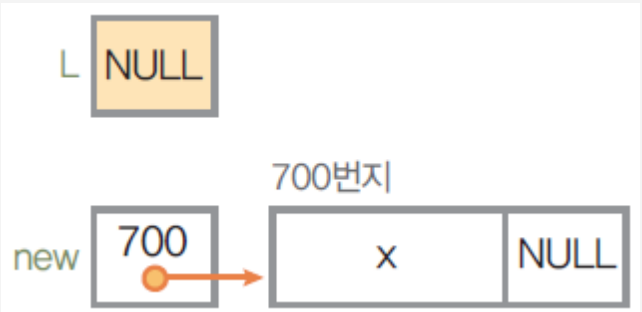
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조 (개정3판), 이지영저, 한빛미디어

5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

- ❶ new ← getNode();
- ❷ new.data ← x;
- ❸ new.link ← NULL;

5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

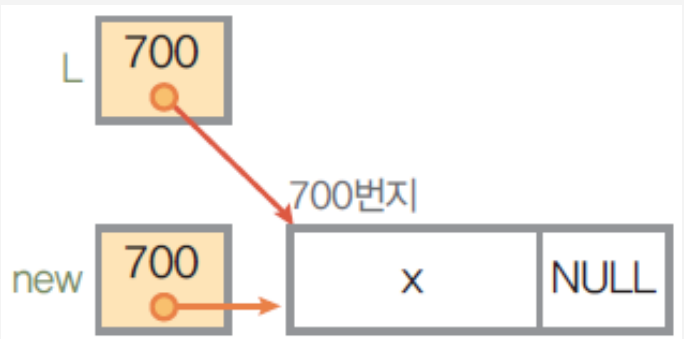
❹ 공백 리스트인 경우



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

5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

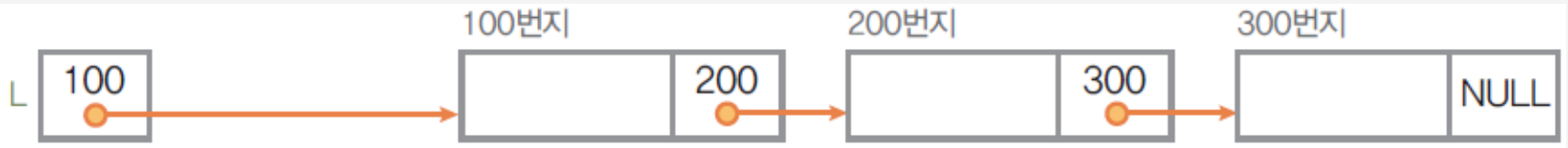
- ❹- a L ← new;
  리스트 포인터 L에 새 노드 new의 주소(700) 저장
  new는 리스트 L의 첫 번째 노드이자 마지막 노드



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

5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

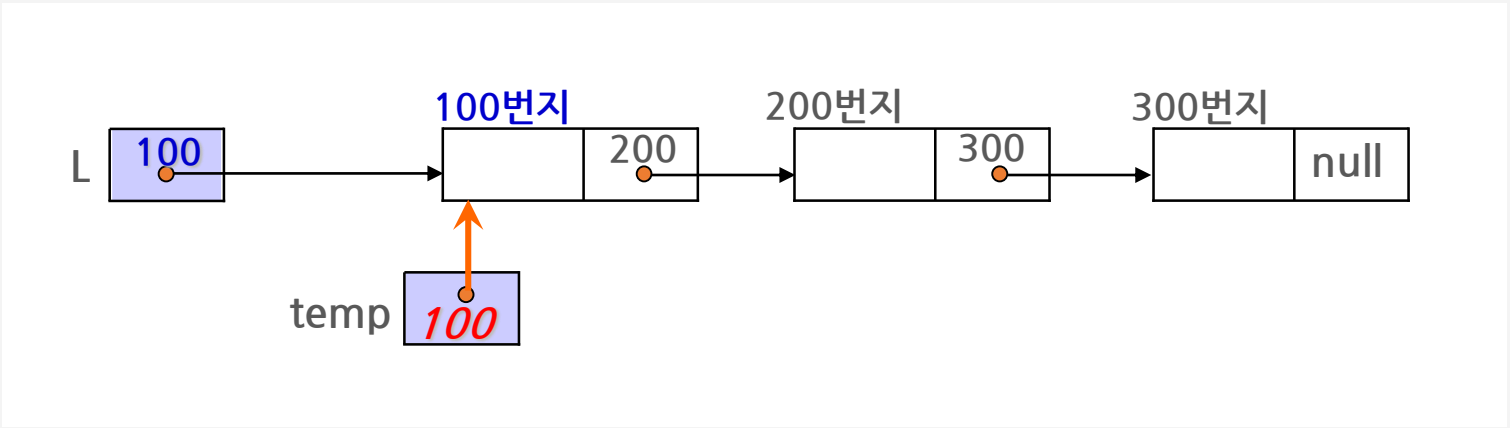
❺ 공백 리스트가 아닌 경우



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

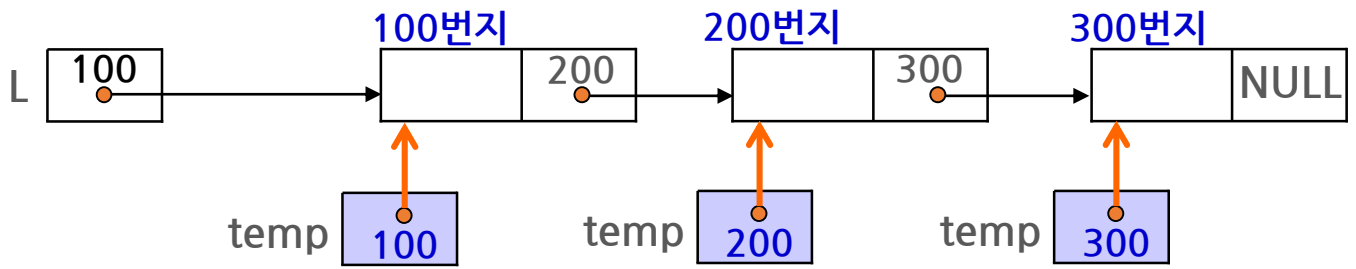
5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

- ❺- a temp ← L;
  현재 리스트 L의 마지막 노드를 찾기 위해서 노드를 **순회할 임시포인터 temp**에 리스트의 **첫 번째 노드의 주소(L)**를 지정



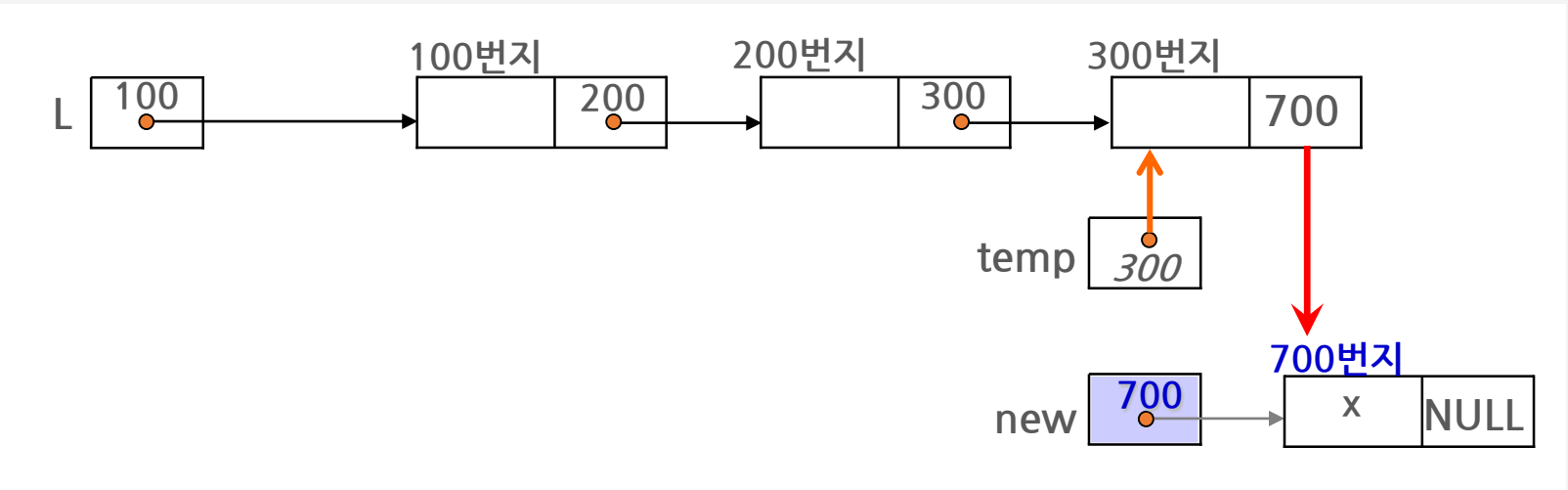
5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

- ❺- b temp ← temp.link;
순회 포인터 temp가 가리키는 노드의 링크 필드가 NULL이 아닌 동안 (while (temp.link ≠ NULL)) 링크 필드를 따라 이동. 링크 필드가 NULL인 노드 즉, 마지막 노드를 찾으면 while 문을 끝내고 ❺- c를 수행



5 단순 연결 리스트의 알고리즘 : 마지막 노드로 삽입하기

- ❺- c temp.link ← new;
  마지막 노드의 링크필드에 새 노드 new 주소 저장



2 단순 연결 리스트의 삭제방법과 알고리즘

1 단순 연결 리스트에서의 삭제 연산

- 철이는 집에 가고 싶다며 기차놀이에서 빠지려 함
(이때 기차 연결이 끊기지 않도록 이름표를 정확히 인수인계 해야 함)

◎ 기차놀이에서 철이가 빠지기 전 상태

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

1 단순 연결 리스트에서의 삭제 연산

- 철이는 집에 가고 싶다며 기차놀이에서 빠지려 함
  (이때 기차 연결이 끊기지 않도록 이름표를 정확히 인수인계 해야 함)

① 앞사람에게 이름표 넘겨주기

② 이름표대로 연결하기

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

2 단순 연결 리스트에서 노드를 삭제하는 방법

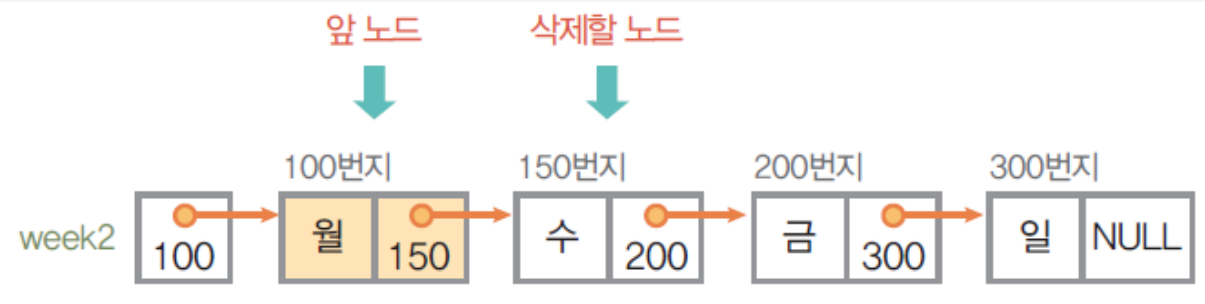
1. 삭제할 노드의 앞 노드를 찾음
2. 앞 노드에 삭제할 노드의 링크 필드값을 저장함
3. 삭제한 노드의 앞 노드와 삭제한 노드의 다음 노드를 연결함

3 단순 연결 리스트 week2=(월, 수, 금, 일)에서 원소 '수' 삭제 과정

▶ 초기 상태

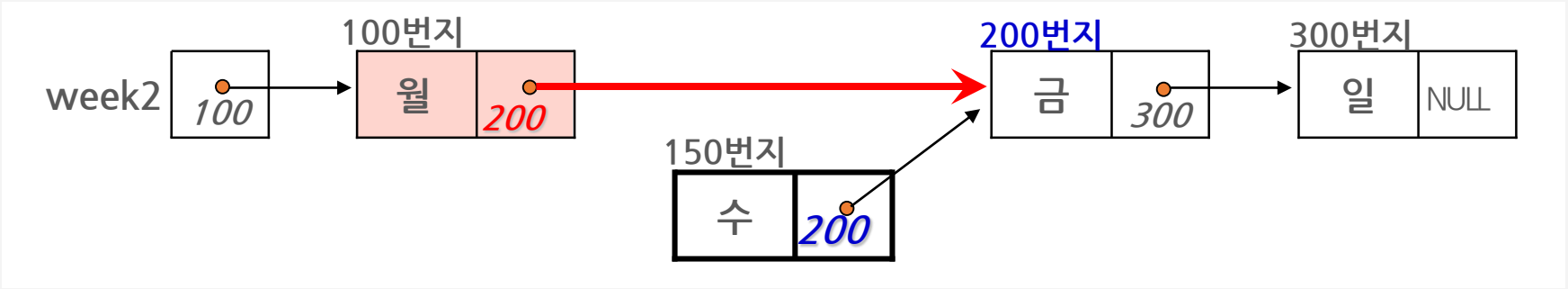
3 단순 연결 리스트 week2=(월, 수, 금, 일)에서 원소 '수' 삭제 과정

❶ 앞 노드를 찾음
: 삭제할 원소의 앞 노드(선행자)를 찾음



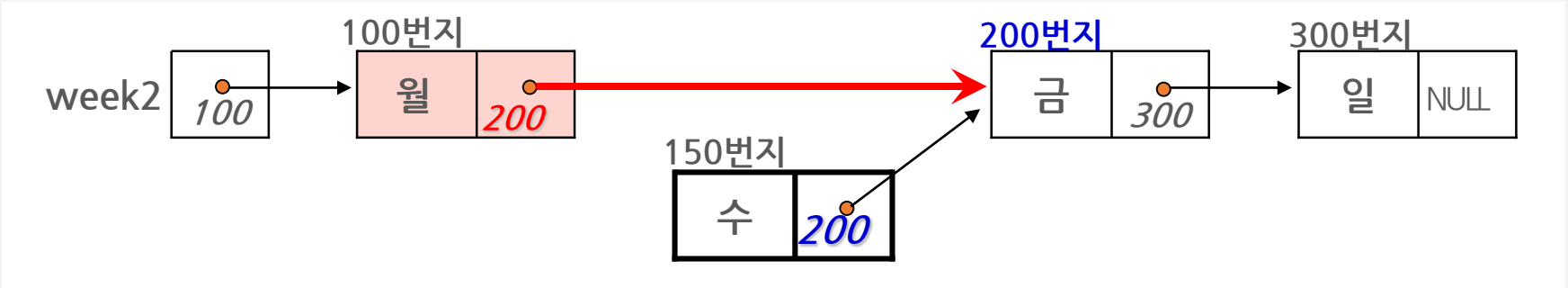
3 단순 연결 리스트 week2=(월, 수, 금, 일)에서 원소 '수' 삭제 과정

❷ 앞 노드에 삭제할 노드의 링크 필드값 저장
: **삭제할 원소 "수"의 링크 필드 값**을 **앞 노드의 링크 필드**에 저장



3 단순 연결 리스트 week2=(월, 수, 금, 일)에서 원소 '수' 삭제 과정

❸ 삭제한 노드의 앞뒤 노드 연결
: 삭제한 노드의 앞 노드인 '월' 노드를 삭제한 노드의 다음 노드인 '금' 노드에 연결



4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

▶ 포인터 old(삭제할 노드)

알고리즘 4-4 단순 연결 리스트의 노드 삭제

```
deleteNode(L, pre)
    ❶ if (L = NULL) then error;
    ❷ else {
        ❷-ⓐ old ← pre.link;
        ❷-ⓑ if (old = NULL) then return;
        ❷-ⓒ pre.link ← old.link;
        ❷-ⓓ returnNode(old);
      }
end deleteNode()
```

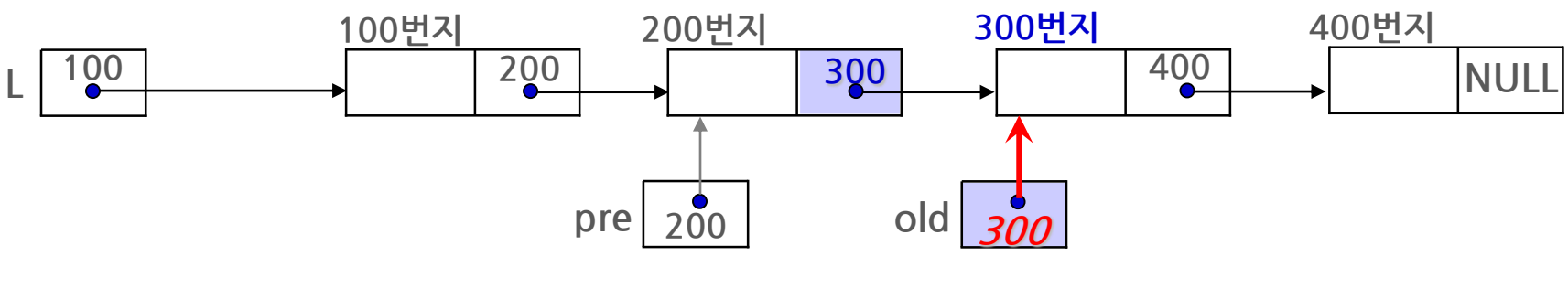
4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

- ❶ 공백 리스트인 경우
  : 공백 연결 리스트 L이 공백이면 오류 처리함

4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

❷ 공백 리스트가 아닌 경우

❷- a old ← pre.link;
노드 pre의 다음노드의 주소(pre.link)를 **포인터 old**에 저장하여, 포인터 old가 다음 노드를 가리키게 함

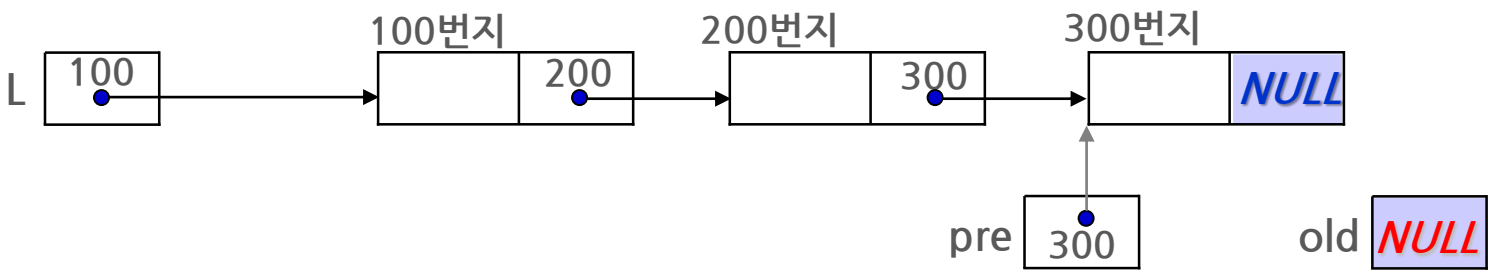


4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

- ❷ 공백 리스트가 아닌 경우

❷- b if (old = NULL) then return;
만약 노드 pre가 리스트의 마지막 노드였다면 :

포인터 pre가 가리키는 노드가 마지막 노드인 경우에
②-ⓐ **old ← pre.link;**
를 수행한 상태



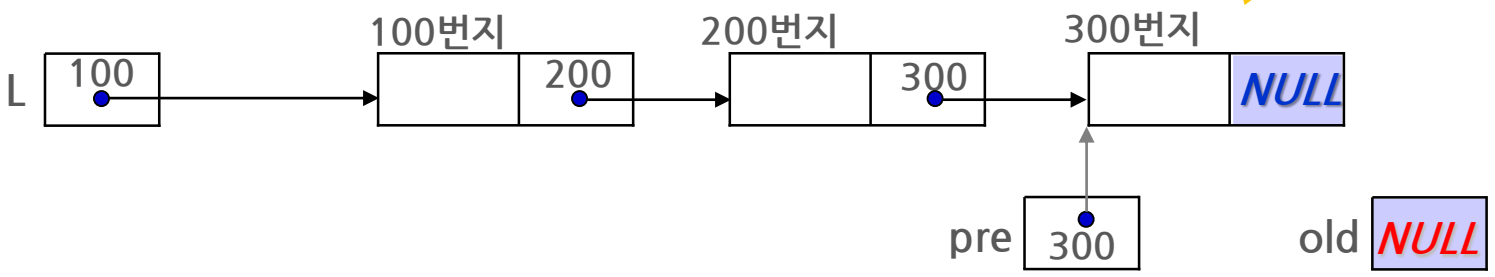
2 | 단순 연결 리스트의 삭제방법과 알고리즘

4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

- ❷ 공백 리스트가 아닌 경우

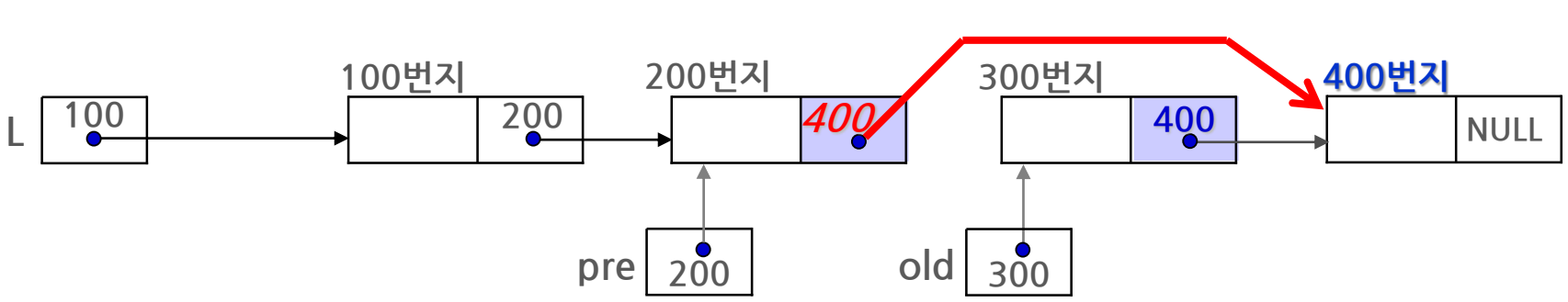
❷- b if (old = NULL) then return;
만약 노드 pre가 리스트의 마지막 노드였다면 :

pre.link의 값은 null이므로 포인터 old의 값은 null이 됨 결국 노드 pre 다음에 삭제할 노드가 없다는 의미이므로 삭제연산을 수행하지 못하고 반환(return)



4 리스트 L에서 포인터 pre가 가리키는 노드의 다음 노드 삭제 알고리즘

❷- c pre.link ← old.link;



❷- d returnNode(old);
삭제한 노드 old의 메모리를 반환

3 단순 연결 리스트의 노드 탐색 알고리즘

3 | 단순 연결 리스트의 노드 탐색 알고리즘

1 노드를 탐색하는 알고리즘과 프로그램

- 연결리스트에서 원소값이 x 인 노드를 탐색하려면
  리스트의 노드를 처음 부터 하나씩 순회하면서
  현재 노드의 데이터 필드값과 x값과 비교하여
  일치하는 노드를 찾아야 함

1 노드를 탐색하는 알고리즘과 프로그램

▶ 노드를 탐색하는 알고리즘
: 리스트의 노드를 순회하기 위해 포인터 temp를 사용함

알고리즘 4-5 단순 연결 리스트의 노드 탐색

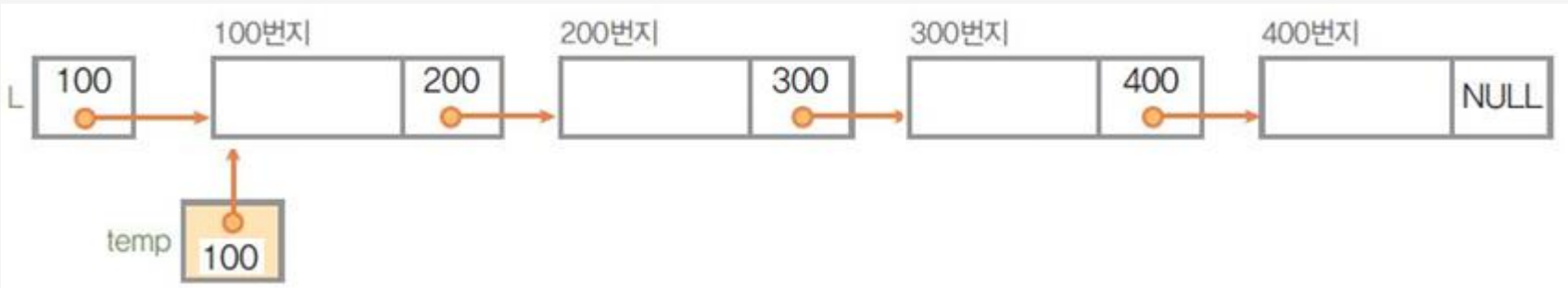
```
searchNode(L, x)
    ❶ temp ← L;
    ❷ while (temp ≠ NULL) do {
          ❷-ⓐ if (temp.data = x) then return temp;
          ❷-ⓑ else temp ← temp.link;
      }
    ❸ return temp;
end searchNode()
```

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

3 | 단순 연결 리스트의 노드 탐색 알고리즘

2 단순 연결 리스트에서 x 노드 탐색 과정

❶ temp ← L;
리스트 L의 시작주소를 노드를 탐색할 때 사용할 순회 포인터 temp에 저장하여 포인터 temp의 시작위치를 지정함

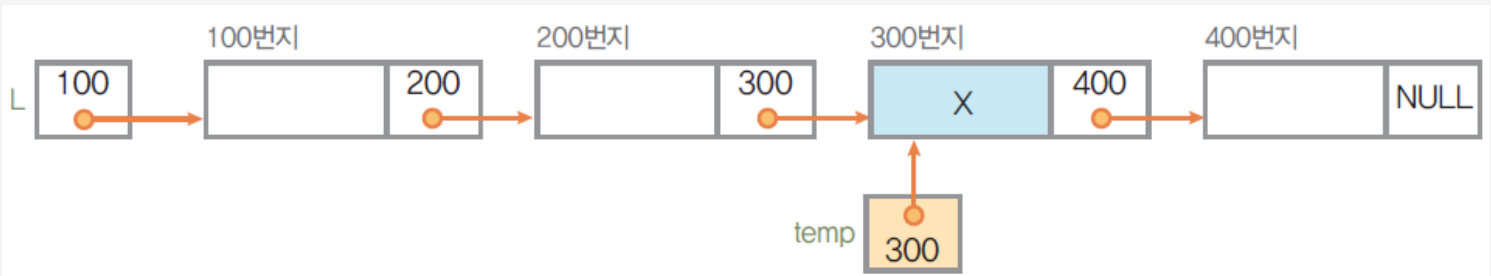


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 단순 연결 리스트에서 x 노드 탐색 과정

❷ 순회 포인터가 NULL이 아닌 경우
(temp가 NULL이 아닌 동안 탐색 연산 반복)

❷- a if (temp.data = x) then return temp;
현재 temp노드의 데이터 필드 값이 탐색 값 x와 같으면 현재 temp값 return하여 x가 있는 노드 주소를 알려줌(이 경우 탐색 성공이 됨)



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

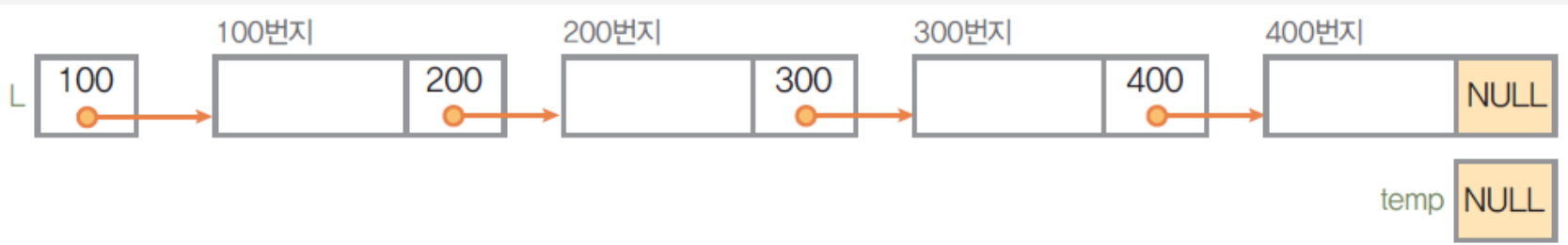
2 단순 연결 리스트에서 x 노드 탐색 과정

- ❷ 순회 포인터가 NULL이 아닌 경우
(temp가 NULL이 아닌 동안 탐색 연산 반복)

  ❷- b else temp ← temp.link;
  현재 필드 값과 탐색 값 x와 같지 않으면 탐색을 계속해야 하므로 링크를 따라 다음 노드로 이동

2 단순 연결 리스트에서 x 노드 탐색 과정

❸ return temp;
While 문에서 x 노드를 찾지 못하고 결국 리스트의 마지막 노드의 링크 필드값인 NULL이 포인터 temp에 설정되어 while문 수행종료
현재 포인터의 temp값인 NULL값을 반환, 탐색값 x가 리스트 L에 없다는 의미가 되므로 탐색 실패



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

3 단순 연결 리스트에서 역순으로 변경하는 과정

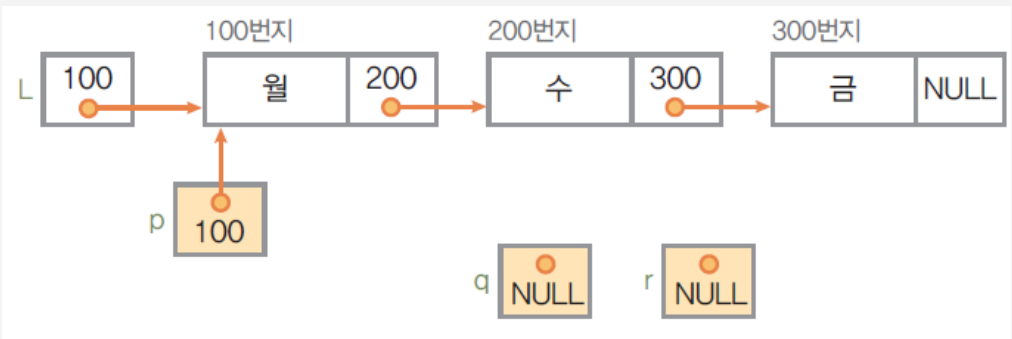
- 리스트의 노드 순서를 역순으로 바꾸는 연산

```
typedef struct ListNode {
   char data[4];
   struct ListNode* link;
} listNode;

typedef struct { //리스트 시작 head 노드
   listNode* head;
} linkedList_h;

void reverse(linkedList_h * L) {
   listNode* p;   listNode* q;   listNode* r;
   p = L->head;  // 포인터 p를 첫 번째 노드에 설정
   q = NULL;   r = NULL;
  // 첫 번째 노드부터 시작해서  노드 간의 연결을 바꿈
   while (p != NULL) {
        r = q;          q = p;
        p = p->link;    q->link = r;
   }
   L->head = q;
}
```

3 단순 연결 리스트에서 역순으로 변경하는 과정

- 시작 상태 : 포인터 p를 첫 번째 노드에 설정

  p = L->head; q = NULL; r = NULL;



- 리스트의 첫 번째 노드부터 링크를 따라 다음 노드로 이동하면서 노드 간의 연결을 반대로 바꿈

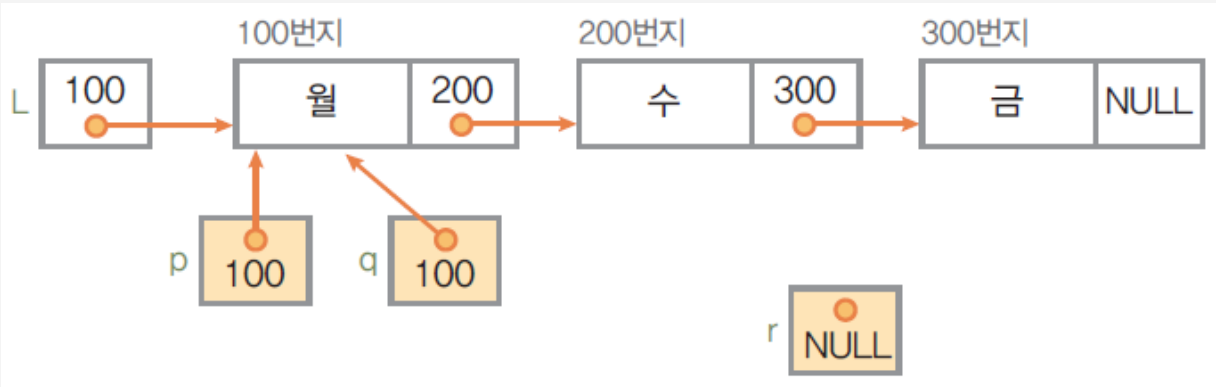
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- 포인터 p가 NULL 값이 아닐 때 까지 반복
  : while (p != NULL)

포인터 q를 포인터 p가 가리키는 현재 노드에 연결
: q = p;



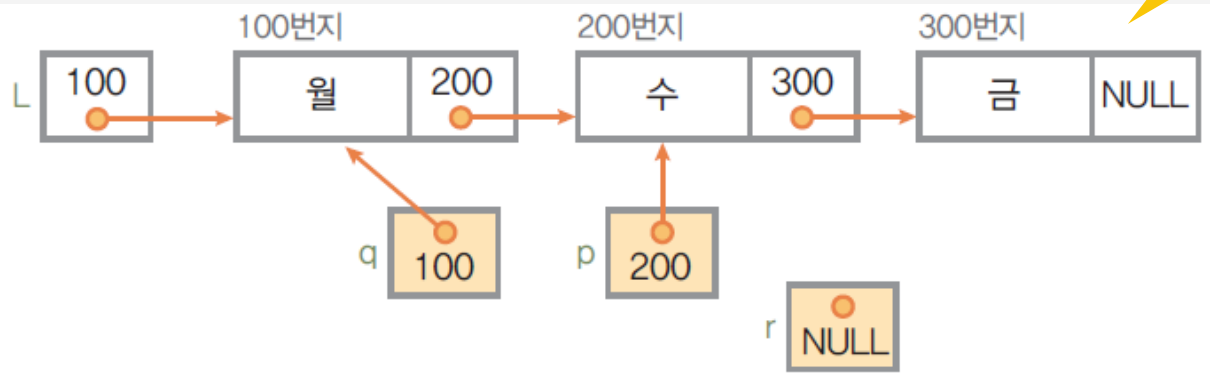
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- 포인터 p가 NULL 값이 아닐 때 까지 반복
  : while (p != NULL)

포인터 p는 링크를 따라 다음 노드로 이동 : p = p->link;



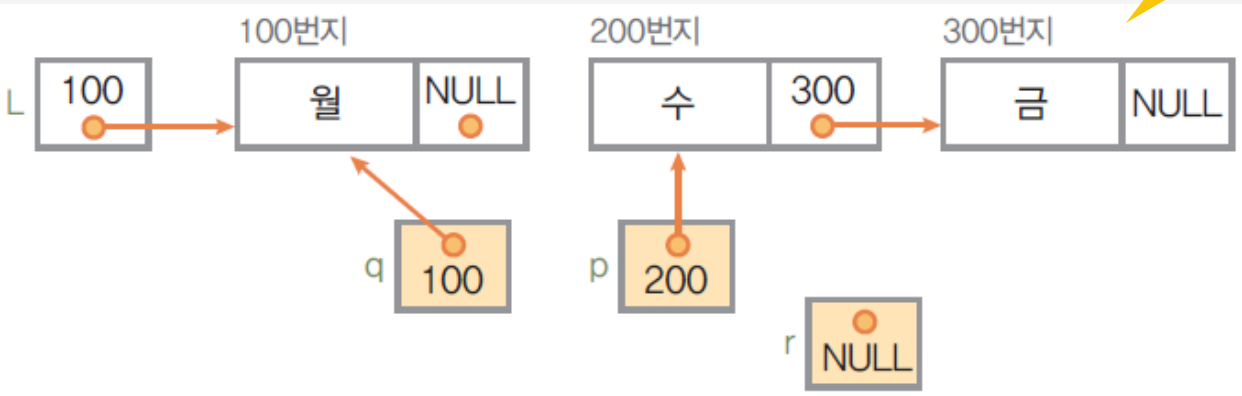
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- 포인터 p가 NULL 값이 아닐 때 까지 반복
  : while (p != NULL)

while 문 한 번 수행 :
q->link = r;



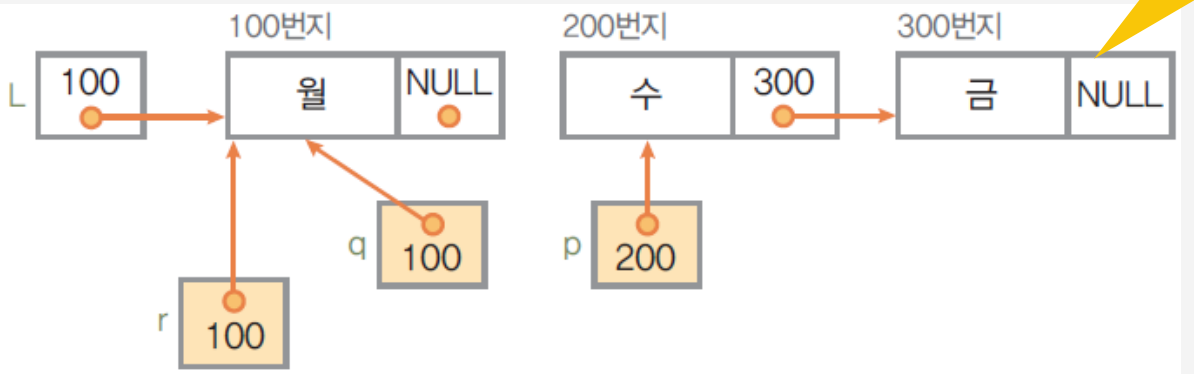

```
* while 구문
while (p != NULL) {
    r = q;
    q = p;
    p = p->link;
    q->link = r;
}
```

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

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행



포인터 r을 포인터 q가 가리키는 노드에 연결: r = q;

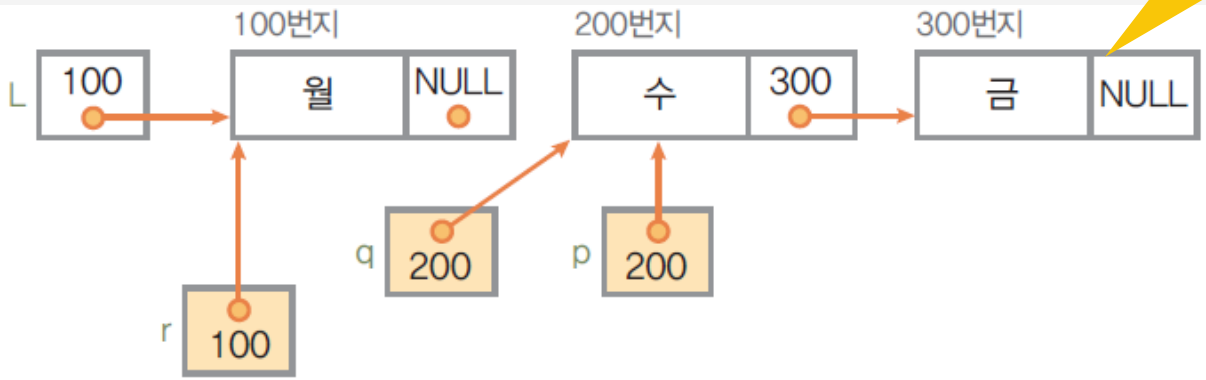
```
* while 구문
while (p != NULL) {
    r = q;
    q = p;
    p = p->link;
    q->link = r;
}
```

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

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행



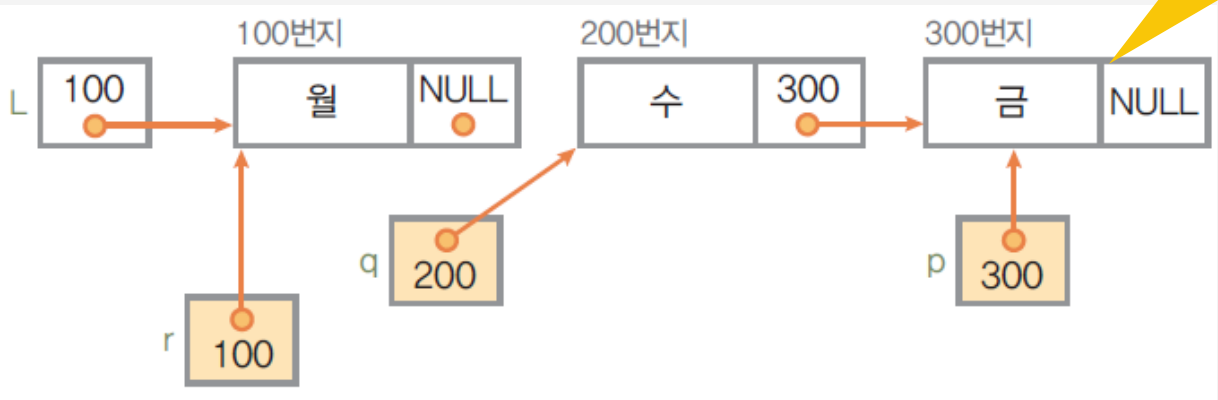
포인터 q를 포인터 p가 가리키는 현재 노드에 연결 : q = p;

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

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행



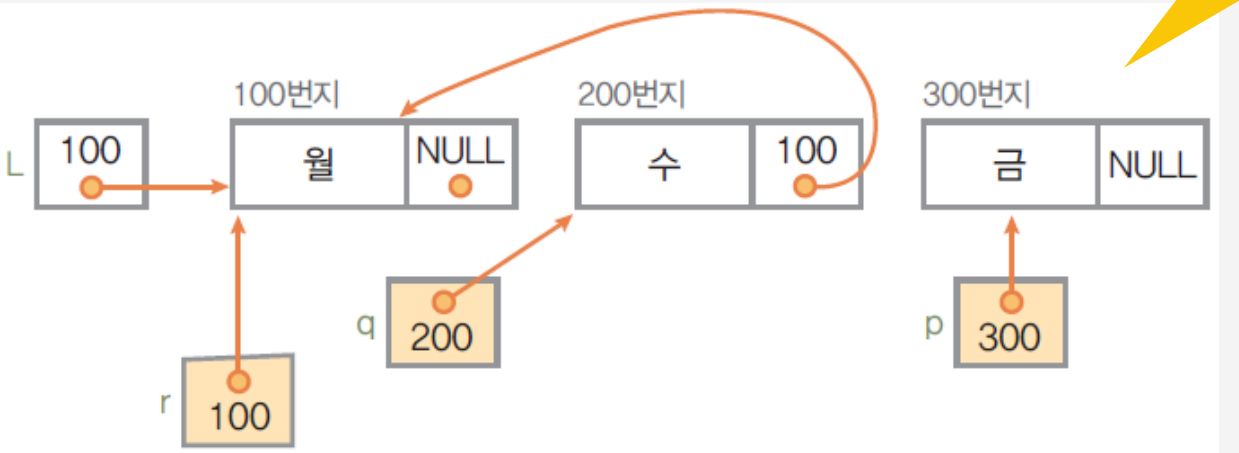
포인터 p는 링크를 따라 다음 노드로 이동 : p = p->link;

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

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행

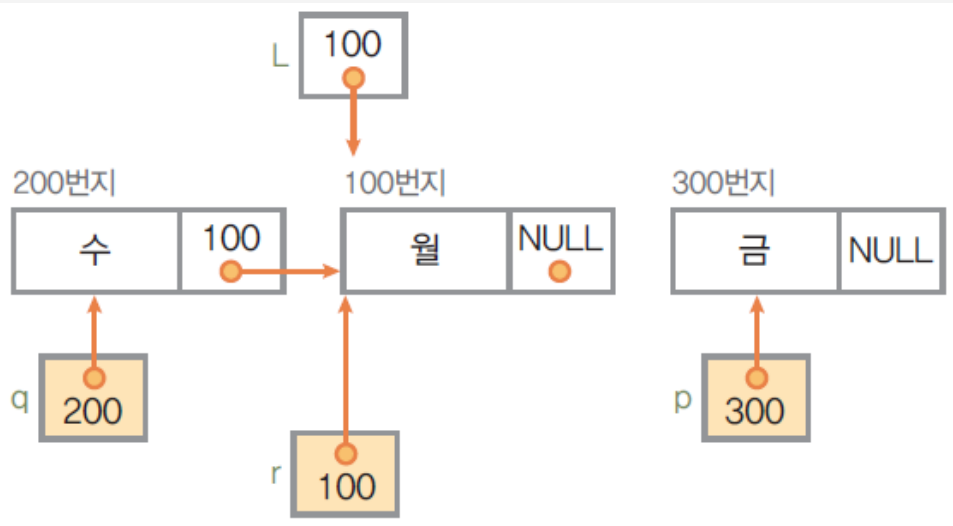


포인터 r의 값(100)을 q 노드의 링크 필드에 설정
: q->link = r;

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

3 단순 연결 리스트에서 역순으로 변경하는 과정

- while 문을 두 번 수행 결과

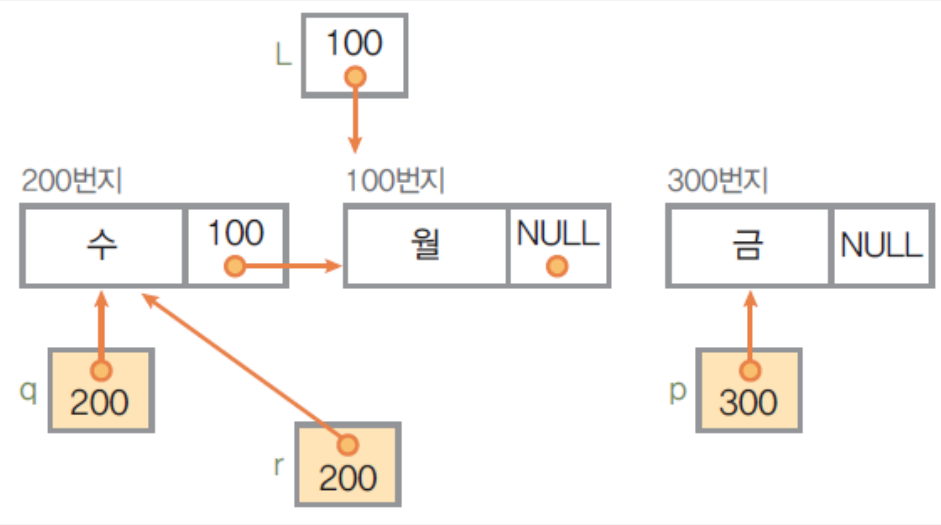


※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행



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

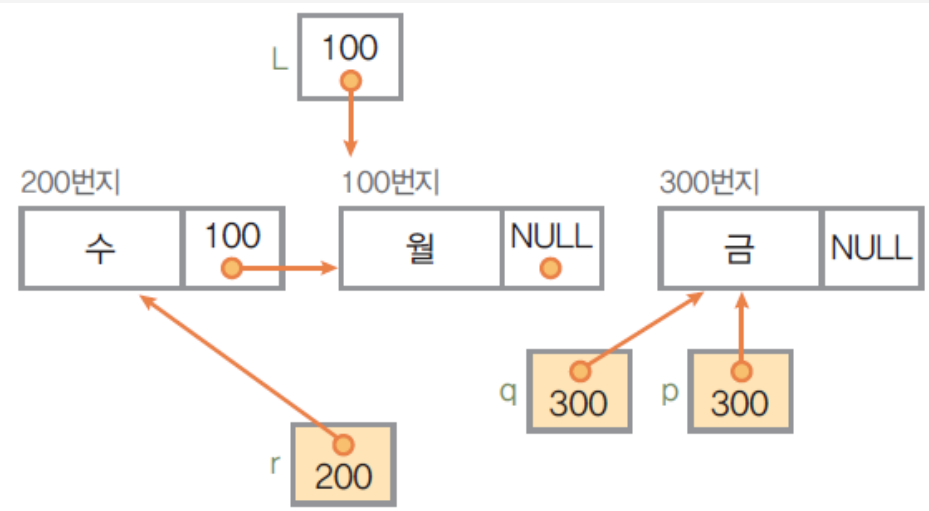
포인터 r을 포인터 q가 가리키는 노드에 연결 : r = q;

```
* while 구문
while (p != NULL) {
    r = q;
    q = p;
    p = p->link;
    q->link = r;
}
```

3 단순 연결 리스트에서 역순으로 변경하는 과정

포인터 q를 포인터 p가 가리키는 노드에 연결 : q = p;

- p != NULL 이므로 while 구문 재실행



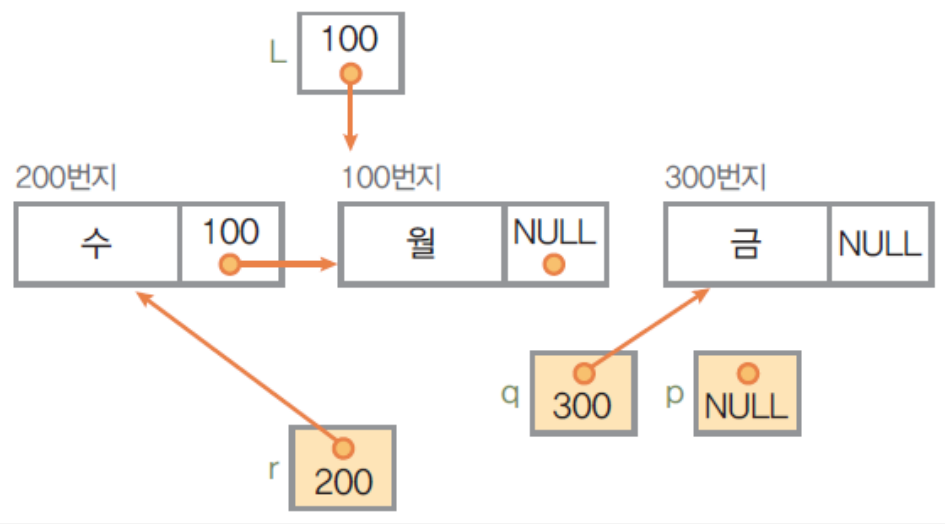
※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 | 단순 연결 리스트의 노드 탐색 알고리즘

3 단순 연결 리스트에서 역순으로 변경하는 과정

포인터 p는 링크를 따라 다음 노드로 이동 : p = p->link;

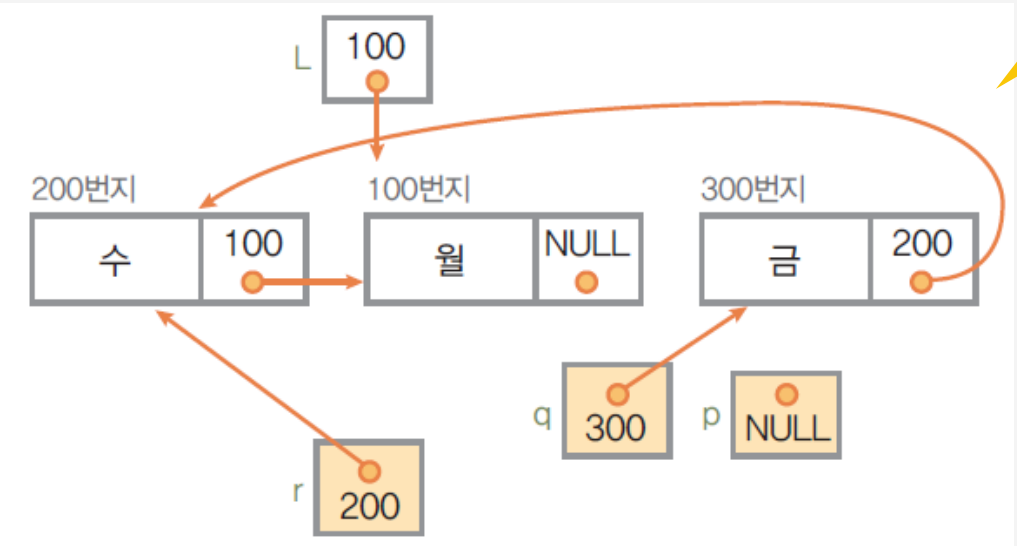
- p != NULL 이므로 while 구문 재실행



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

3 단순 연결 리스트에서 역순으로 변경하는 과정

- p != NULL 이므로 while 구문 재실행

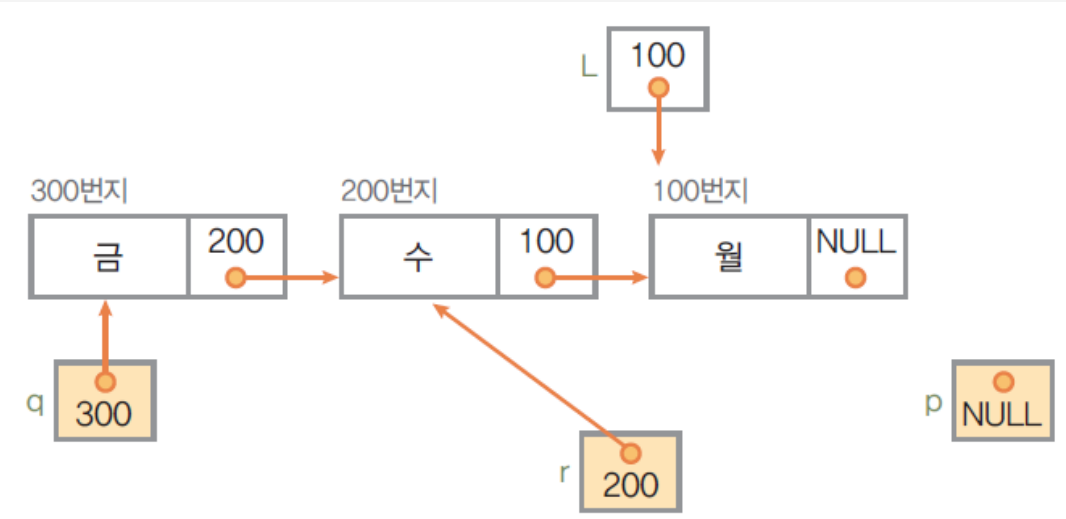


포인터 r의 값(200)을 q 노드의 링크 필드에 설정
: q->link = r;

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

3 단순 연결 리스트에서 역순으로 변경하는 과정

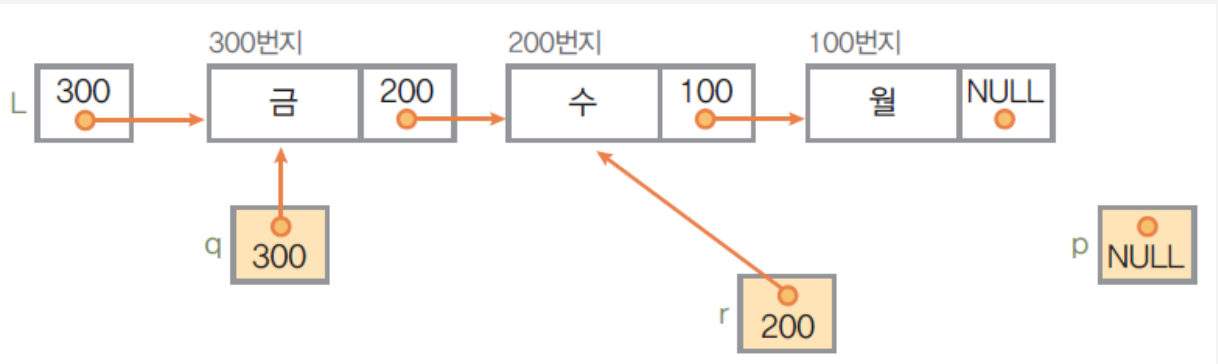
- while 문을 세 번 수행 결과



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

3 단순 연결 리스트에서 역순으로 변경하는 과정

- 완성된 연결 리스트 : L->head = q;



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