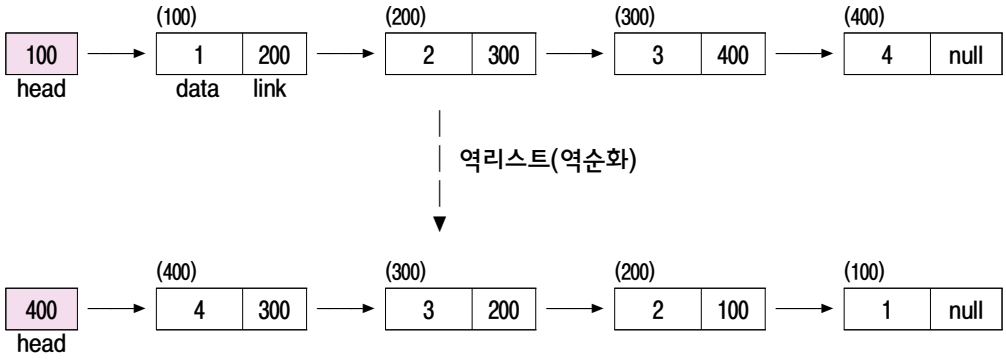
● 역리스트

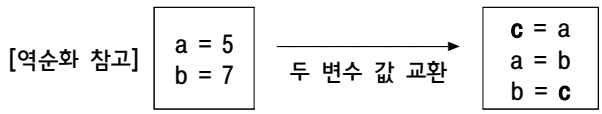
단순연결리스트에 대한 **역리스트(역순화)**에 대해서 살펴본다.



다음은 단순연결리스트를 역순화 하는 함수이다.

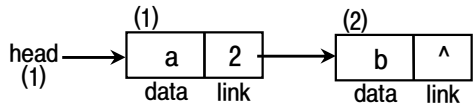
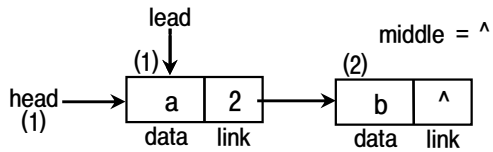
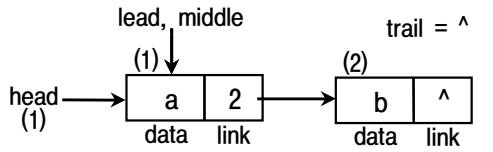
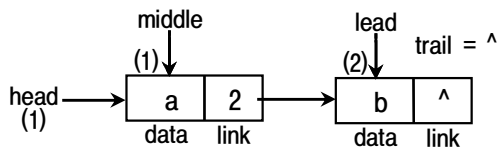
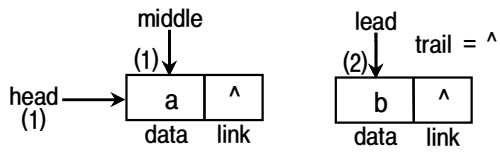
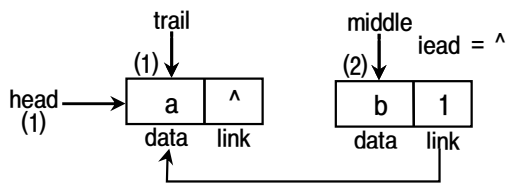
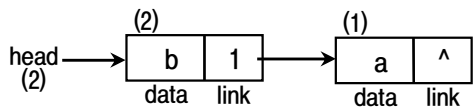
```
node *invert_list(node *head)
{
  node *lead, *middle, *trail;  //3개의 포인터 선언
  lead = head;                  //역순으로 만들 리스트를 가리킴
  middle = NULL;                //역순으로 만든 리스트를 가리킴
  while(lead){
    trail = middle;             //trail은 middle을 뒤 따른다
    middle = lead;              //middle은 lead를 뒤 따른다
    lead = lead->link;          //lead는 다음 노드를 추적해간다
    middle->link = trail;       //middle->link는 이전 노드를 가리키게 한다
  }
  return middle;                //역순으로 만든 리스트 반환
}
```

- 리스트 역순화는 오름차순으로 된 자료를 내림차순으로 바꾸는데 사용될 수 있다.
- 리스트 역순화 알고리즘은 두 변수의 값을 교환하는 원리와 비슷한 측면이 있다.



2 http://cafe.daum.net/pass365(홍재연)

// 시험장에서 리스트 역순화는 다음처럼 2개의 노드를 이용하여 검증하면 된다.(어렵게 출제되면)

<table>
<tr><td>초기상태</td><td> 
</td><td>역순화하기 전</td></tr>
<tr><td>반복하기 전</td><td> 
</td><td>lead = head;
middle = NULL;</td></tr>
<tr><td rowspan="3">첫 번째 반복</td><td> 
</td><td rowspan="3">while(lead){
trail = middle;
middle = lead;
lead = lead->link;
middle->link = trail;
}</td></tr>
<tr><td> 
</td></tr>
<tr><td> 
</td></tr>
<tr><td>두 번째 반복</td><td> 
</td><td>while(lead){
trail = middle;
middle = lead;
lead = lead->link;
middle->link = trail;
}</td></tr>
<tr><td>역리스트</td><td> 
</td><td>return middle;</td></tr>
</table>

- 리스트 역순화가 잘 이해되지 않는다.라는 **질문**을 많이 받는다.
- 아마, **아인슈타인**이 보아도 그냥 잘 이해되지 않을 것 같다.
- 리스트 역순화 알고리즘 길이는 짧지만 결코 쉬운 것이 아니다. **천재**들이 할 수 있는 것이다.
- 우리같은 **범인**들은 천재들이 만들어 놓은 것을 **이해**할 수 있도록 하면 된다.
- 천재가 만들어 놓은 것을 **눈**으로 그냥 보고 **이해**할 수 있으면 천재와 **비슷**하다.
- **공부**하면서 **손**으로 직접 그려보지 않고 이해되지 않는다.라는 말은 그만하자.(직접 그려보면 이해된다)
- 적어도 손으로 그려가면서 이해할 수 있어야 전산**7**급에 **합격**할 수 있지 않을까?

기출문제 분석

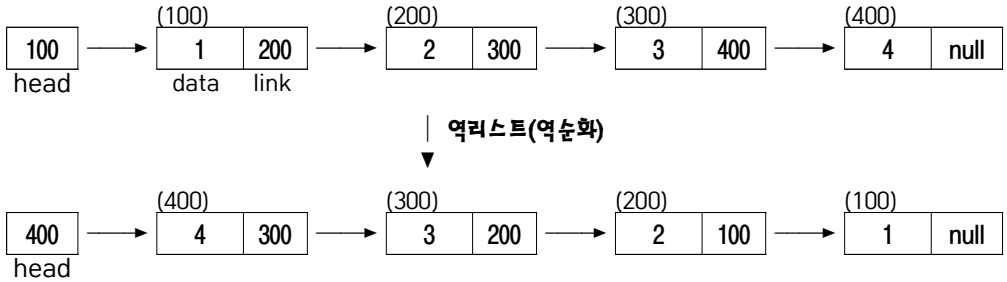
1. 아래의 알고리즘은 주어진 단순연결리스트를 역순으로 변환하는 알고리즘이다. 알고리즘의 ㉠에 들어갈 내용으로 옳은 것은?(단, 리스트의 시작 주소를 나타내는 포인터는 start이며 노드의 연결 포인터 필드는 link이다) [2010년 국가 7급] [2014년 국가 7급]

```
p = start;
q = NULL;
while(p != NULL){
  [     ㉠     ]
  p = p->link;
  q->link = r;
}
start = q;
```

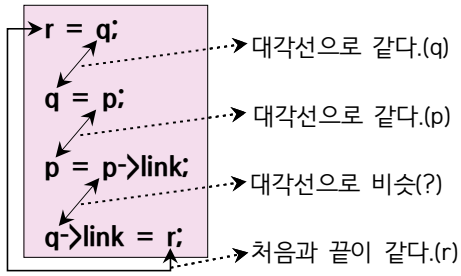
① q = q->link; r = q;　　② r = q; q = p;

③ r = q; q = p->link;　　④ q = q->link;

♤ 단순연결리스트에 대한 역리스트(역순화) – 3, 4년 주기로 출제됨



◈ 단순연결리스트에 대한 역리스트 문제 푸는 방법



- 좌측에, 역리스트 문제 푸는 방법을 제시하였다.
- 무슨 말인지? 이해할 수 있을 것이다.
- 만약, 좌측에 제시한 방법 이외의 방법으로 풀면 1분 안에 풀기가 어려울 것이다.
- 최종적으로 공무원 시험은 시간 싸움이다.

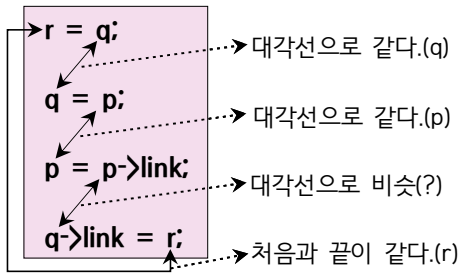
정답 : ②

2. 다음은 단순연결리스트(singly linked list) L에서 리스트의 원소를 역순으로 바꾸는 함수이다. 함수에서 ㉠ 부분에 들어갈 프로그램 코드(code)로 옳은 것은? [2019년 국가 7급]

```
------------------------------
typedef struct node{
   struct node *link;
   char data;
} listNode;
typedef struct{
   listNode *head;
} linkedList;
void reverse(linkedList *L) {
   listNode *p, *q, *r;
   p = L->head;
   q = r = NULL;
   while ( p != NULL)  {
      r = q;
      q = p;
      p = p->link;
        ㉠
   }
   L->head = q;
}
------------------------------
```

① q = q->link;　　② q->link = r;
③ p->link = q;　　④ r->link = q;

♤ 단순연결리스트에 대한 역리스트 문제 푸는 방법



- 좌측에,
  역리스트 문제 푸는 방법을 제시하였다.
- 무슨 말인지? 이해할 수 있을 것이다.
- 만약, 좌측에 제시한 방법 이외의 방법으로 풀면 1분 안에 풀기가 어려울 것이다.
- 최종적으로 공무원 시험은 시간 싸움이다.

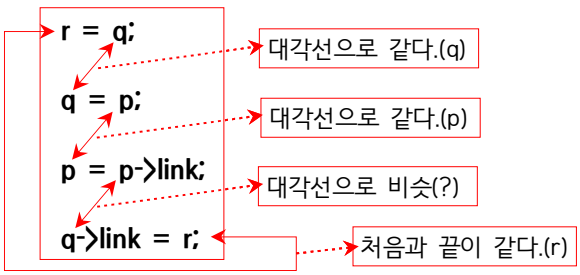
정답 : ②

3. 〈보기〉는 단순연결리스트(singly linked list)에서 원소들의 순서를 역순으로 변환하는 C 언어 프로그램이다. (가), (나), (다)에 들어갈 내용을 옳게 짝지은 것은? [2020년 서울 7급]

```
----<보기>----------------------
typedef struct node {
  int key;
  struct node *link;
} ListNode;
ListNode *reverse(ListNode *L)
{
  ListNode *p, *q, *r;
  p = L;
  q = NULL;
  r = NULL;
  while (p != NULL) {
      ____(가)____;
      ____(나)____;
      p = p→link;
      ____(다)____;
  }
  return q;
}
---------------------------------
```

	(가)	(나)	(다)
①	q = p	r = q	q→link = r
②	r = q	q = p	q→link = r
③	r = q	q = p	r = q→link
④	q = p	r = q	r = q→link

♤ 단순연결리스트에 대한 역리스트 문제 푸는 방법



- 좌측에, 역리스트 문제 푸는 방법을 제시하였다.
- 무슨 말인지? 이해할 수 있을 것이다.
- 만약, 좌측에 제시한 방법 이외의 방법으로 풀면 1분 안에 풀기가 어려울 것이다.
- 최종적으로 공무원 시험은 시간 싸움이다.

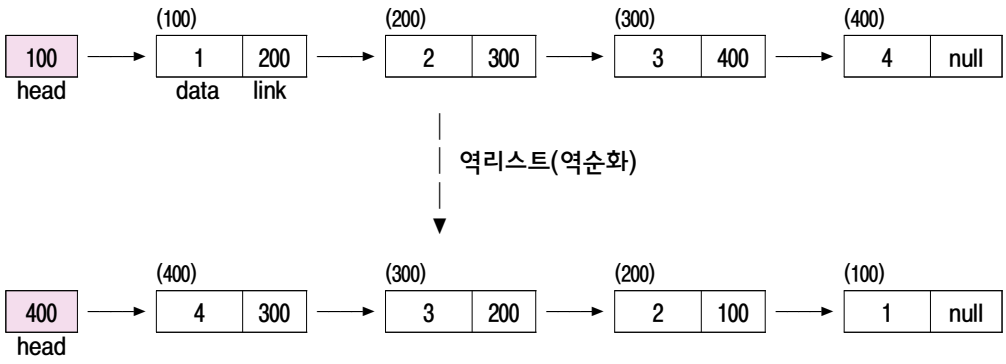
정답 : ②

4. 다음 C 코드의 reverseList()는 단순 연결리스트에 있는 노드의 순서를 역순으로 만드는 함수이다. (가)~(다)에 들어갈 내용을 바르게 연결한 것은? [2022년 국가 7급]

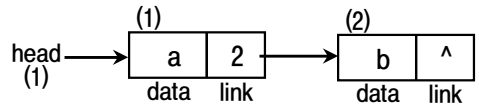
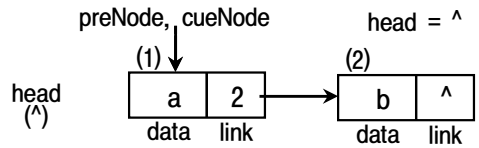
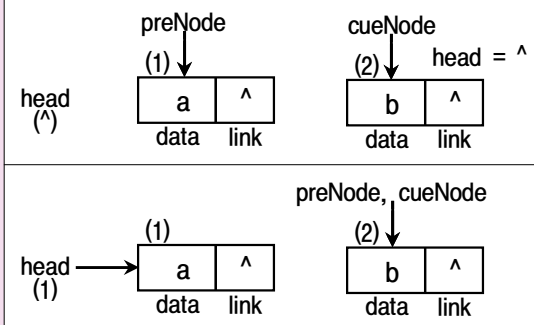
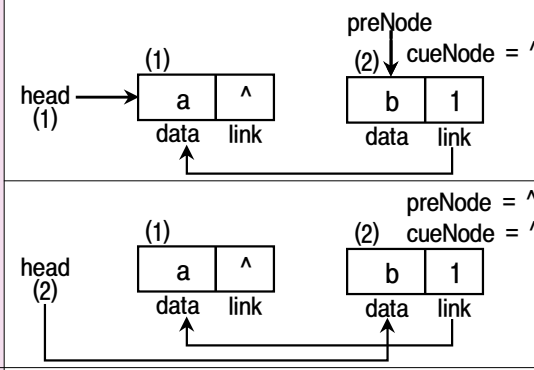
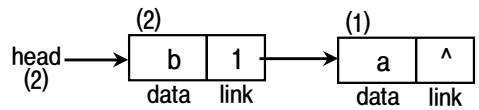
```
struct node {
  int data;
  struct node *link;
}*head;
void reverseList() {
  struct node *prevNode, *curNode;
  prevNode = head;
  curNode = head;
  head = NULL;
  while ( (가) ) {
      curNode = curNode->link;
      prevNode->link = head;
      (나) ;
      (다) ;
  }
}
```

	(가)	(나)	(다)
①	curNode != NULL	head = curNode	curNode = prevNode
②	curNode != NULL	head = prevNode	prevNode = curNode
③	head != NULL	head = curNode	curNode = prevNode
④	head != NULL	head = prevNode	prevNode = curNode

♤ 역리스트



// 시험장에서 리스트 역순화는 다음처럼 2개의 노드를 이용하여 직접 그림을 그리면서 검증하면 된다.

<table>
<tr><td>초기상태</td><td> 
</td><td>역순화하기 전</td></tr>
<tr><td>반복하기 전</td><td> 
</td><td>prevNode = head;
curNode = head;
head = NULL;</td></tr>
<tr><td>첫 번째 반복</td><td> 
</td><td>while((가) curNode != NULL)
{
curNode = curNode->link;
prevNode->link = head;
(나) head = prevNode ;
(다) prevNode = curNode ;
}</td></tr>
<tr><td>두 번째 반복</td><td> 
</td><td>while((가) curNode != NULL)
{
curNode = curNode->link;
prevNode->link = head;
(나) head = prevNode ;
(다) prevNode = curNode ;
}</td></tr>
<tr><td>역리스트</td><td> 
</td><td>return middle;</td></tr>
</table>

- 리스트 역순화가 잘 이해되지 않는다.라는 **질문**을 많이 받는다.
- 아마, **아인슈타인**이 보아도 그냥 잘 이해되지 않을 것 같다.
- 리스트 역순화 알고리즘 길이는 짧지만 결코 쉬운 것이 아니다. **천재**들이 할 수 있는 것이다.
- 우리같은 **범인**들은 천재들이 만들어 놓은 것을 **이해**할 수 있도록 하면 된다.
- 천재가 만들어 놓은 것을 **눈**으로 그냥 보고 **이해**할 수 있으면 천재와 **비슷**하다.
- **공부**하면서 **손**으로 직접 그려보지 않고 이해되지 않는다.라는 말은 그만하자.(직접 그려보면 이해된다)
- 적어도 손으로 그려가면서 이해할 수 있어야 전산7급에 **합격**할 수 있지 않을까?

정답 : ②