

※ 다음의 제약조건을 위배하지 않고 프로그램을 작성하라. 특히 ListNode와 ListType은 변경할 수 없다.
문제의 해결에 필요한 함수는 ADT 기본 함수 포함 본인이 직접 작성한다. 또한 아래의 번호는 시뮬레이션의 순서를 의미하며 답안을 제출할 때에는 전체 코드만 제출하면 된다. 문제 번호에 맞춰서 코드를 분리하여 제출하지 말라는 의미임.

1. 단순연결리스트를 사용한다. 이때 헤드 노드, 헤드 포인터 중 원하는 하나를 사용하여 구현하며, 테일 노드와 size 변수 등은 만들 수 없다.

<pre>typedef int element; typedef struct ListNode { element data; struct ListNode *next; }ListNode;</pre>	<pre>void init(ListType *L) { //헤드 노드인 경우 L->H = (ListNode*)malloc(sizeof(ListNode)); L->H->next = NULL; }</pre>
<pre>typedef struct ListType { ListNode *H; }ListType;</pre>	<pre>void init(ListType *L) { //헤드 포인터인 경우 L->H = NULL; }</pre>

2. 이 연결리스트로 정수 리스트 하나(L1)와 빈 리스트 하나(L2)를 만든다. 정수 리스트의 정수 데이터는 반드시 1 ~ 20까지의 난수 15개를 발생시켜 생성한다. 본인이 입력함수(scanf())를 이용하여 임의로 값을 입력하는 경우 총 점수의 50%가 감점이다. 리스트 출력 함수에서 마지막에 NULL이 출력되도록 한다.

<pre>ListType L1, L2; init(&L1); init(&L2); srand(time(NULL)); for(int i = 0; i < 15; i++) insertFirst(&L1, rand() % 20 + 1); print(&L1);</pre>	<pre>10 7 16 17 8 12 5 20 13 19 17 17 16 11 10 NULL</pre> <p>==> 생성된 리스트</p>
---	---

3. 정수 리스트(L1)를 위치(pos)를 입력받아 두 개의 리스트(L1, L2)로 나누는 partition() 함수를 작성하고 테스트한다. pos까지가 L1 리스트이고 pos 다음부터 마지막까지가 L2 리스트이다.

<pre>int pos; scanf("%d", &pos); partition(&L1, &L2, pos); print(&L1); print(&L2);</pre>	<pre>5 10 7 16 17 8 NULL 12 5 20 13 19 17 17 16 11 10 NULL</pre> <p>==> 분할된 두 리스트</p>
--	--

4. 위에서 분할된 두 리스트 L1, L2를 L1으로 다시 연결하는 concat() 함수를 작성하고 테스트한다. L1은 원래의 리스트로 복원될 것이고 L2는 위에서 분할된 결과 그대로일 것이다.

<pre>concat(&L1, &L2); print(&L1); print(&L2);</pre>	<pre>10 7 16 17 8 12 5 20 13 19 17 17 16 11 10 NULL 12 5 20 13 19 17 17 16 11 10 NULL</pre> <p>==> 다시 복원된 L1</p>
--	---

5. 리스트에서 특정한 값을 모두 삭제하는 deleteNode() 함수를 작성하고 테스트한다. 삭제할 값은 시뮬레이션 리스트에 가장 많이 중복된 값으로 선택한다.

<pre>element e; scanf("%d", &e); deleteNode(&L1, e); print(&L1); print(&L2);</pre>	<pre>17 10 7 16 8 12 5 20 13 19 16 11 10 NULL 12 5 20 13 19 16 11 10 NULL</pre> <p>==> 노드 17이 모두 삭제됨</p>
--	---

6. 리스트를 역순으로 만드는 reverse() 함수를 만들고 테스트한다. 주어진 환경에서 이를 구현하려면 세 개의 포인터 p, q, r이 필요하다. 리스트 L2를 역순으로 만들어보자.

<pre>reverse(&L2); print(&L2);</pre>	<pre>10 11 16 19 13 20 5 12 NULL</pre> <p>==> 역순으로 바뀐 L2</p>
--	---

7. 리스트의 모든 노드를 삭제하는 deleteAll() 함수를 만들고 테스트하라.

<pre>deleteAll(&L1); print(&L1);</pre>	<pre>NULL</pre> <p>L1이 삭제되어 NULL만 출력</p>
--	--