



- 아래와 같이 실행되도록 main()함수 구성

```
E:\LectureW[2020-1]W[2020-1] 데이터구조론\Src\LinkedLists.... - □ ×
(1)공백리스트 생성하기
L=()
리스트에 저장된 데이터 개수: 0

(2)리스트에 10, 50, 80 노드를 순서대로 삽입하기
L=(10, 50, 80)
리스트에 저장된 데이터 개수: 3

(3)리스트에서 50 노드 탐색하기
50 노드를 찾았습니다

(4)50 노드 뒤에 60 노드 삽입하기
L=(10, 50, 60, 80)
리스트에 저장된 데이터 개수: 4

(5)리스트에서 80 노드 삭제하기
노드 삭제 성공!
L=(10, 50, 60)
리스트에 저장된 데이터 개수: 3

-----
Process exited after 0.3784 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

- Linked List ADT의 모든 연산 구현
- SLinkedList.h 제공, SLinkedList.c 및 SLinkedListMain.c 완성

- 리스트의 노드 순서를 역순으로 바꾸는 **reverse()** 연산 추가
- 아래와 같이 실행되도록 **main()**함수 구성

```
E:\LectureW[2020-1]\W[2020-1] 데이터구조론\Src\LinkedList2.exe
(1)공백리스트 생성하기
L=()
리스트에 저장된 데이터 개수: 0

(2)리스트에 월, 화, 목 노드를 순서대로 삽입하기
L=(월, 화, 목)
리스트에 저장된 데이터 개수: 3

(3)목 노드 뒤에 일 노드 삽입하기
목 노드를 찾았습니다.
L=(월, 화, 목, 일)
리스트에 저장된 데이터 개수: 4

(4)리스트에서 수 노드 탐색하기
찾는 데이터가 없습니다.

(5)리스트 순서를 역순으로 바꾸기
L=(일, 목, 화, 월)
리스트에 저장된 데이터 개수: 4

-----
Process exited after 0.04217 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

- SLinkedList2.h 제공, **SLinkedList2.c** 및 **SLinkedListMain2.c** 완성
 - element의 data type의 변경으로 인한 **SLinkedList2.h** 부분 수정 필요

- 리스트에 노드 삽입 시 오름차순으로 정렬되어 들어가도록 **sortInsert()** 연산 추가
- 아래와 같이 실행되도록 main()함수 구성

```
E:\Lecture\HW[2020-1]\HW[2020-1] 데이터구조론\Src\LinkedList3.exe
(1)공백리스트 생성하기
L=()
리스트에 저장된 데이터 개수: 0

(2)리스트에 10, 80, 50 노드를 삽입하되 오름차순으로 들어가도록 하기
L=(10)
L=(10, 80)
L=(10, 50, 80)
리스트에 저장된 데이터 개수: 3

(3)리스트에서 50 노드 탐색하기
50 노드를 찾았습니다

(4)40 노드 삽입하기(오름차순)
L=(10, 40, 50, 80)
리스트에 저장된 데이터 개수: 4

(5)리스트에서 80 노드 삭제하기
노드 삭제 성공!
L=(10, 40, 50)
리스트에 저장된 데이터 개수: 3

-----
Process exited after 0.02708 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

- SLinkedList3.h 제공, **SLinkedList3.c** 및 **SLinkedListMain3.c** 완성