Data Structures

Lab # 07

Data & Knowledge Engineering Laboratory
Kyung Hee University, Korea

Chapter 5 Programming Exercise

실습 준비

1번 문제

- ...₩labplus_CRLF₩labplus₩Lab, C++ 3rd₩Chapter5₩ListLL₩SListLL의 SortedType.h 파일을 사용합니다.
- SortedType.h 파일 내의 InsertItem과 DeleteItem 함수 구현부를 삭제합니다.
- ...₩labplus_CRLF₩labplus₩Lab, C++ 3rd₩Chapter6₩Functions from Text₩circle.cpp 파일의 struct NodeType을 제외한 내용을 SortedType.h에 붙여넣습니다.

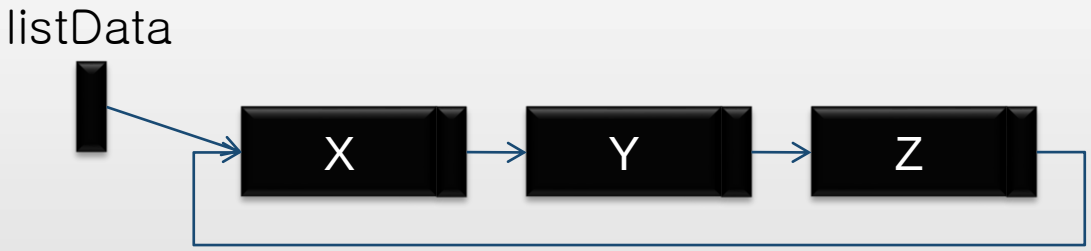
2번 문제

- ...₩labplus_CRLF₩labplus₩Lab, C++ 3rd₩Chapter5₩stackLL의 StackType.h와 StackType.cpp를 사용
- chap06.pdf 42 페이지에 소개된 Copy Constructor를 참고하여 Deep Copy 적용

Exercise 1

- **정렬 리스트 자료구조의 모든 원소를 역순으로 출력하는 PrintReverse 함수를 작성하시오.**
  - 멤버 함수로 구현
    - 프로토타입 : void SortedType<ItemType>::PrintReverse()

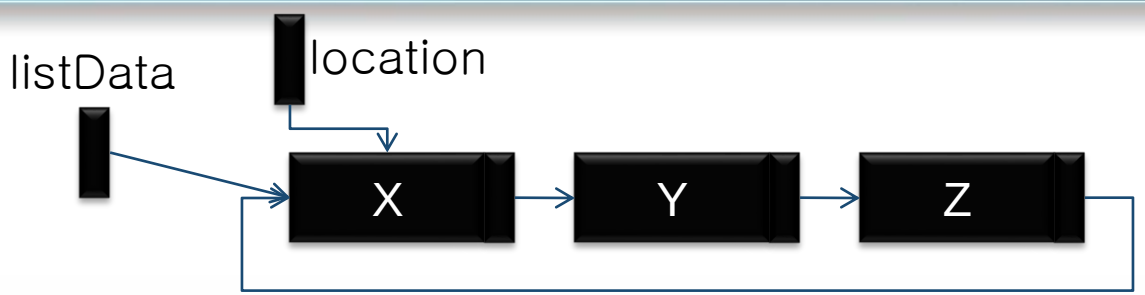
- **예제**



  - 1. 입력된 데이터의 순서가 X,Y,Z라고 가정할 때, PrintReverse 함수로 출력되는 결과는 Z,Y,X 이다. (입력된 데이터의 역순으로 출력)
  - 2. 순환 리스트의 특성을 고려하여 PrintReverse() 함수를 작성할 것

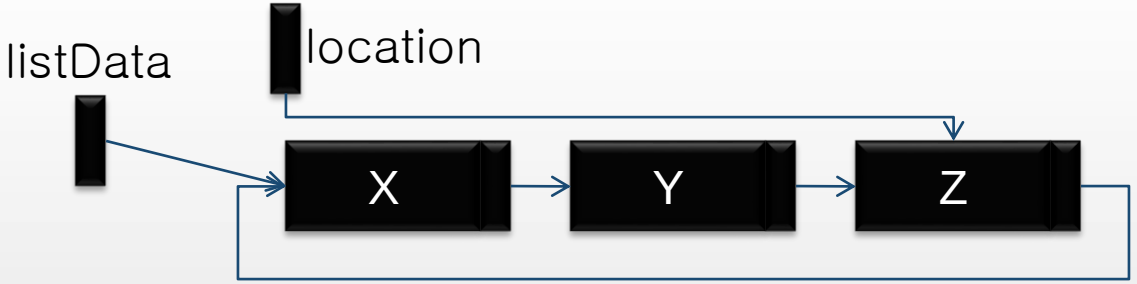
1-help slide

Solution.1



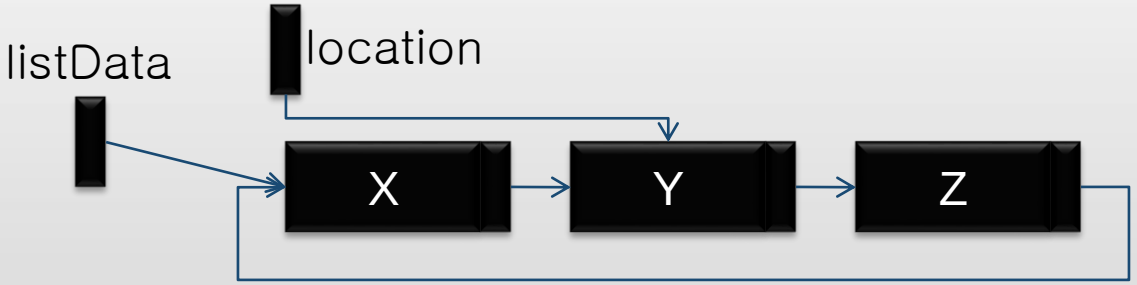
초기상태
location
= listData;

Z 출력



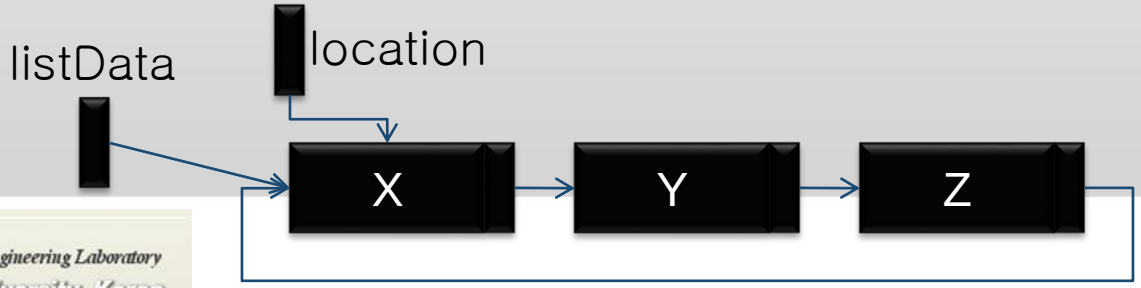
Location을
length-1만큼 이동
Z 출력.

Y 출력



Location을
length-2만큼 이동
Y 출력.

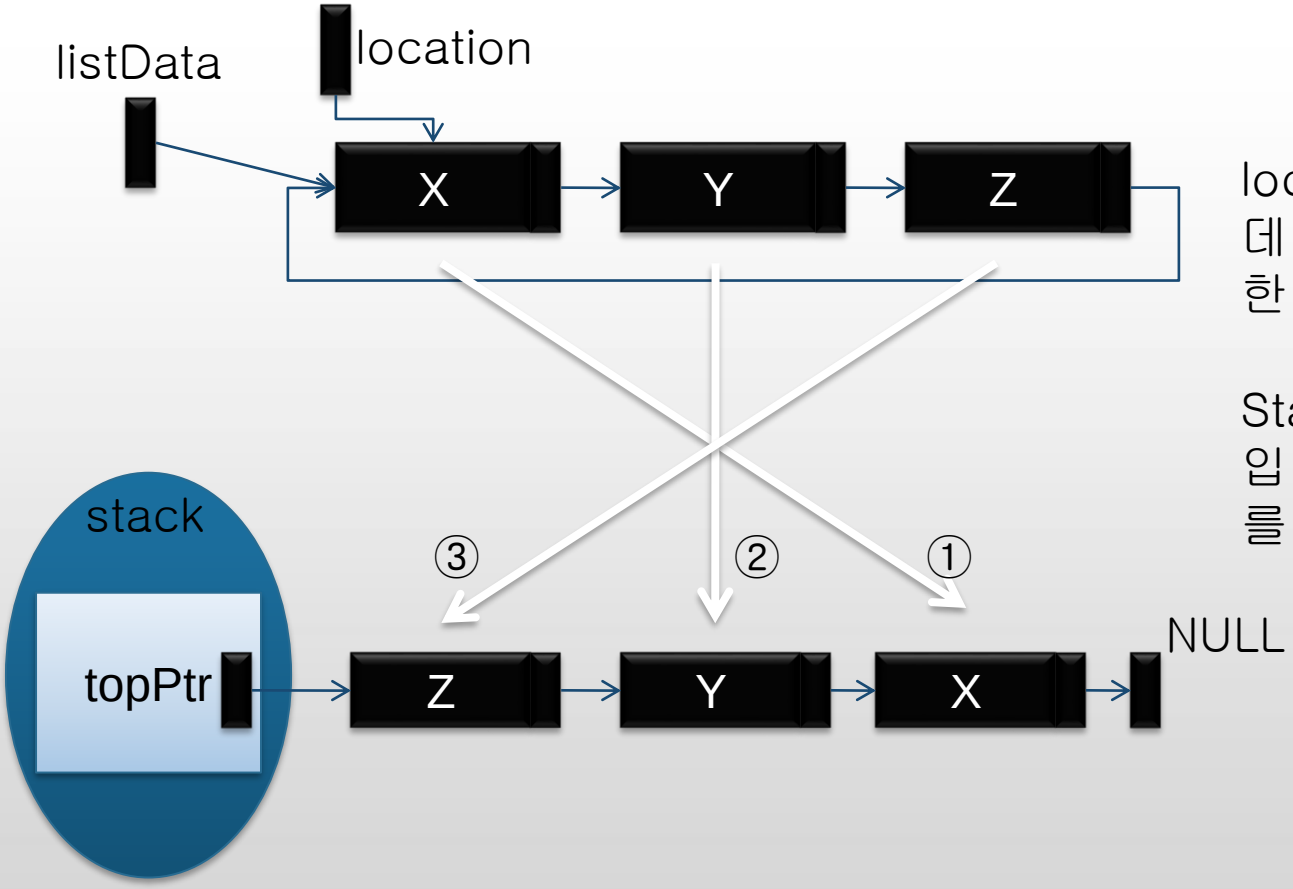
X 출력



Location을
length-3만큼 이동
X 출력.

1-help slide

Solution.2



location이 이동하면서 데이터를 stack에 저장한다.

Stack은 LIFO이므로, 입력의 역순으로 데이터를 출력한다

1-help slide

※프로그램을 종료 할 때 “_block_type_is_valid”와 같은 에러가 나타날경우 ~SortedType() 부분을 수정합니다.

원문

```
template <class ItemType>
SortedType<ItemType>::~SortedType()
// Post: List is empty; all items have been deallocated.
{
    NodeType<ItemType>* tempPtr;
    while (listData != NULL)
    {
        tempPtr = listData;
        listData = listData->next;
        delete tempPtr;
    }
}
```

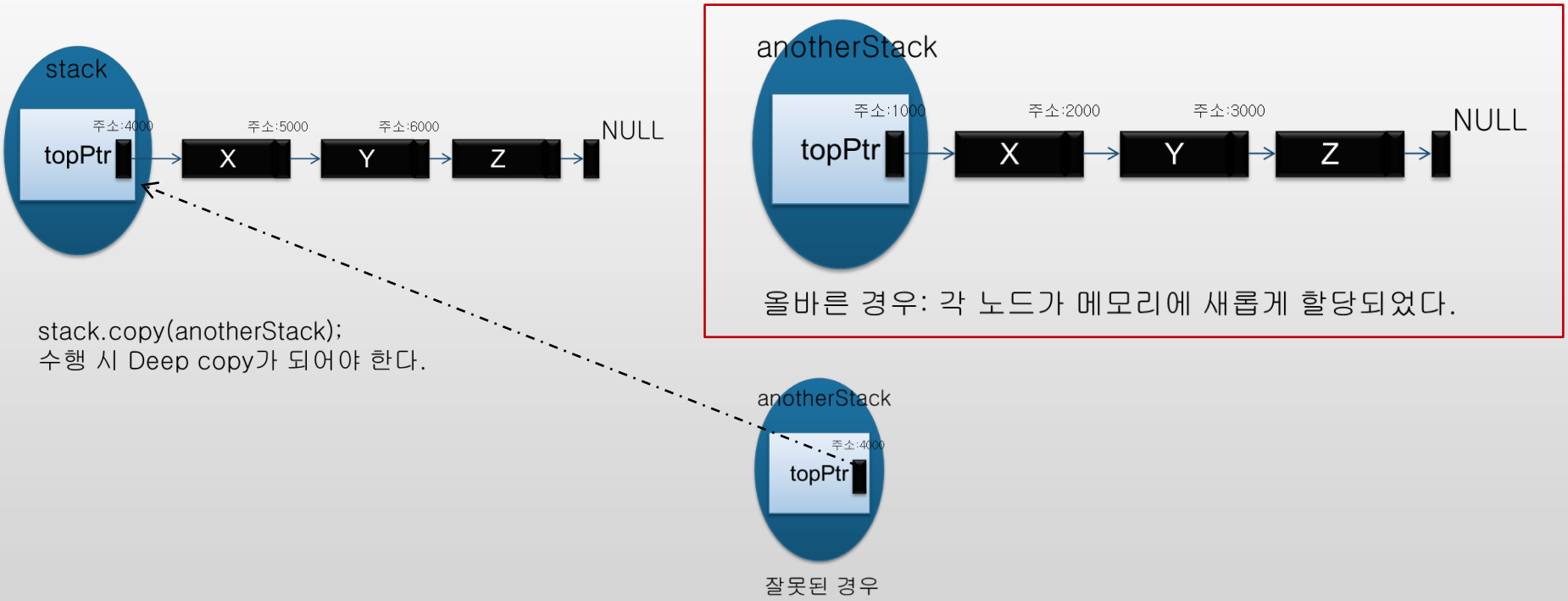
```
while (listData != NULL)
```

```
int length = LengthIs();
for(int i=0; i<length; i++)
```

Exercise 2

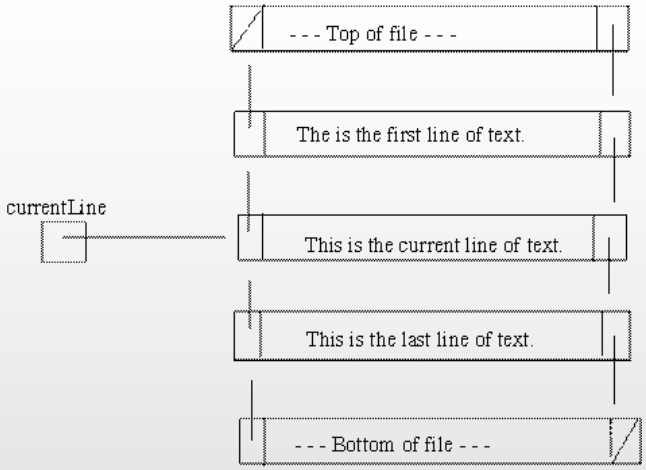
- **Stack의 모든 원소를 새로운 스택에 복사하는 Copy함수를 구현하시오.**
  - Shallow Copy가 아닌 Deep Copy로 원소를 복사해야함
  - 프로토타입 : void StackType::Copy(StackType& anotherStack)

- **예제**



3. Exercise 11 (1/2)

- **Double Linked List를 사용하여 다음과 같은 구조를 가지는 TextEditor 클래스를 구현하시오.**
  - 노드 한 개에서 문자열의 최대 길이는 80으로 제한

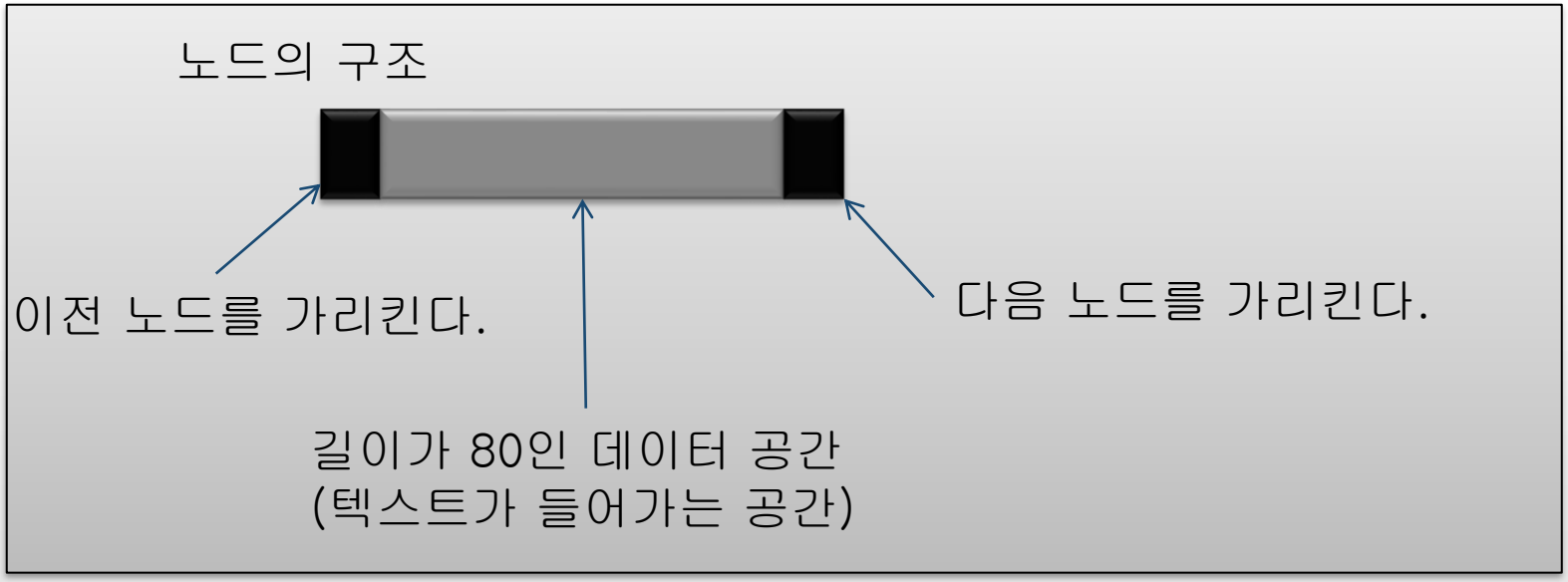


  - a. 위의 구조에 대한 노드 타입을 정의하시오.
  - b. 클래스 생성자를 구현하세요

3. Exercise11 (2/2)

- c. GoToTop함수와 GoToBottom를 작성하시오.
  - GoToTop()
    - Funtion : Goes to top of the list
    - Postcondition : currentLine is set to access the first line of text
  - GoToBottom()
    - Function : Goes to bottom of the list
    - Postcondition : currentLine is set to access the last line of text
- d. c의 Big-O를 생각해보고, 위 함수가 O(1)이 되는 경우를 서술하시오.
- e. InsertItem함수를 작성하시오.
  - InsertLine(char* newline)
    - Function : Inserts newLine at the current Line
    - Postcondtions : newLine has been inserted after current-Line. currentLine points to newLine

■ **A. 위의 구조에서 노드 타입을 정의하시오.**



위의 3가지 요소를 가지는 노드를 정의하세요.

```
struct LineType
{
   …
}
```

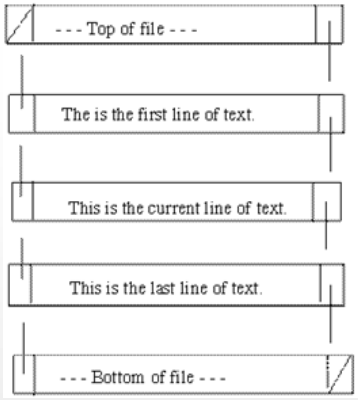
B. 클래스 생성자를 구현하세요.

```
TextEditor::TextEditor()
{
•Header 정의
•Trailer 정의
•위 두 노드를 주어진 문장으로 초기화
•두 노드의 back과 next에 알맞은 값 대입
•LineType *currentLine이 header를 가리킨다.
…
}
```

생성자는 위와 같은 내용을 가지고 있어야 합니다.

C. GoToTop함수와 GoToBottom를 작성하시오.

❖ GoToTop() 실행 후



← CurrentLine

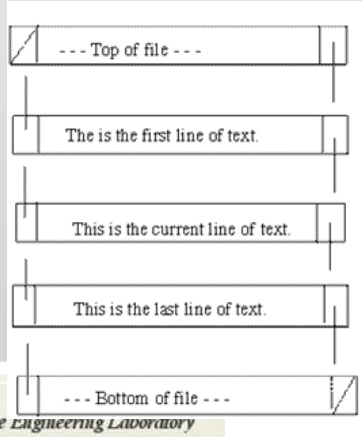
* line이 하나도 없는 경우 trailer를 가리킴

처음과 끝으로 currentLine을 이동시킬때 while()문을 이용하세요.

예시

```
GoToTop( )
{           while(currentLine의 뒤의 노드가 NULL값이 아닌 경우)
                  currentLine을 뒤로 이동시킨다.
            ... //나머지 구현
}
```

❖ GoToBottom() 실행 후



← CurrentLine

* line이 하나도 없는 경우 header를 가리킴

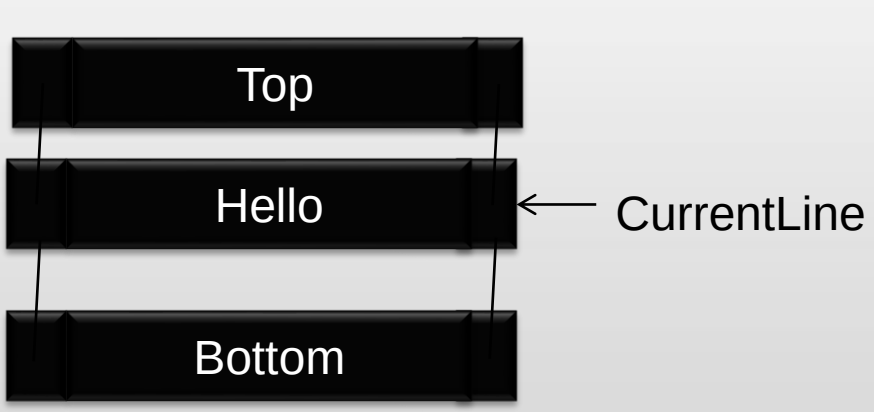
3-help slides

- **d. c의 Big-O를 생각해보고, 위 함수가 O(1)이 되는 경우를 생각하세요.**
  - 구조에 header와 tailer가 있다는 것과 노드가 앞,뒤를 가리킨다는 것을 참고하세요.

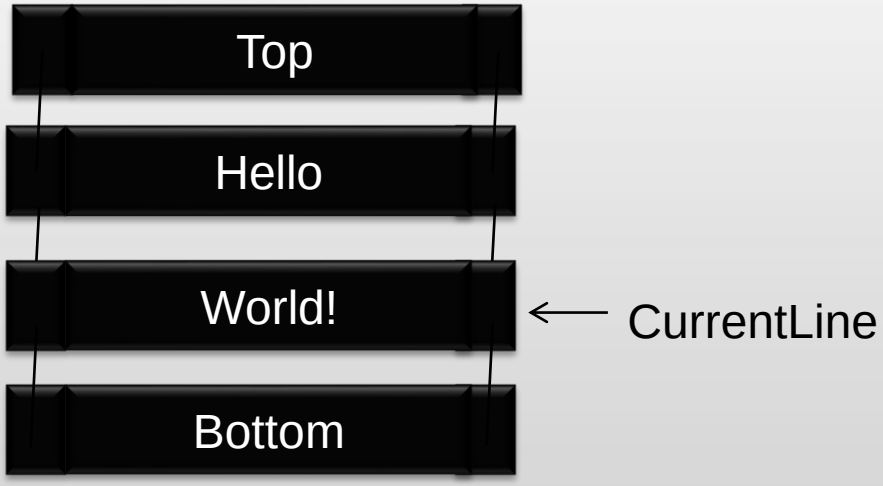
header의 다음 노드가 첫 노드입니다.
tailer의 전 노드가 끝 노드 입니다.

- **E. InsertItem함수를 작성하시오.**

예제)



InsertItem 호출 후



현재 CurrentLine 다음에 새로운 노드를 삽입하고 CurrentLine을 이동

* CurrentLine이 trailer를 가리키는 경우에는 trailer 앞에 삽입