

Data Structures

Lab # 07



■ 1번 문제

- ❖ ...\\labplus_CRLF\\labplus\\Lab, C++ 3rd\\Chapter5\\ListLL\\SListLL
Linked List의 베이스를 SortedType.h 파일을 사용합니다.
- ❖ SortedType.h 파일 내의 InsertItem과 DeleteItem을 삭제합니다.
- ❖ ...\\labplus_CRLF\\labplus\\Lab, C++ 3rd\\Chapter6\\Functions from
Text\\circle.cpp 파일을 열어서 노드정의 부분을 제외한 내용을 SortedType.h
에 붙여넣습니다.

■ 2번 문제

- ❖ 샘플 코드는 Lab6에서 사용했던 "StackLL"을 사용하여 copy 멤버 함수를 추가
하세요.
(...\\labplus\\Lab, C++ 3rd\\Chapter5\\stackLL)
- ❖ 강의노트 P.42에 소개된 CopyConstructor 코드를 참고하여 Deep Copy를 적
용하세요.

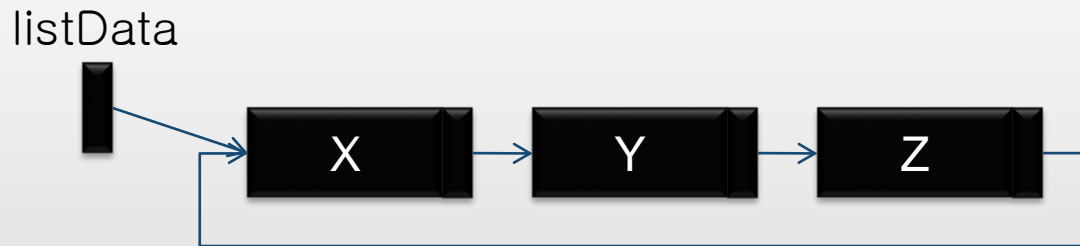
1. Exercise 4

■ 정렬 리스트 자료구조 내의 모든 원소를 역순으로 출력하는 PrintReverse 함수를 작성하시오.

❖ 멤버 함수로 구현

- 프로토타입 : `void SortedType<ItemType>::PrintReverse()`

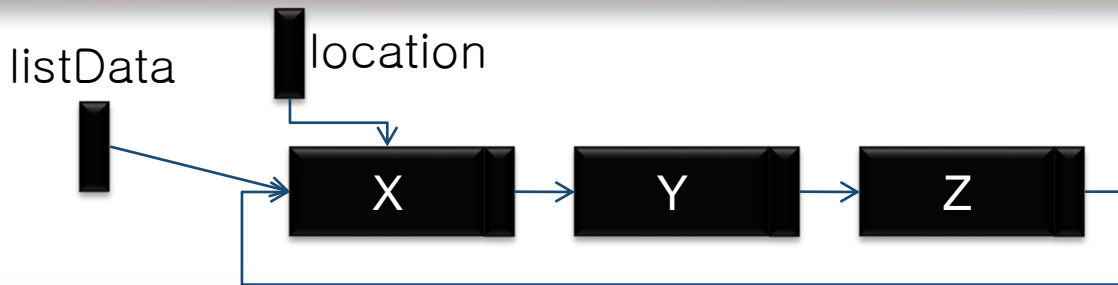
■ 예제



- ❖ 1. 입력된 데이터가 X,Y,Z라고 가정할 때, PrintReverse 함수에 의하여 출력되는 결과는 Z,Y,X 이다. (입력된 데이터의 역순으로 출력되어야함)
- ❖ 2. 순환 리스트의 특성을 고려하여 "PrintReverse()"를 작성할 것

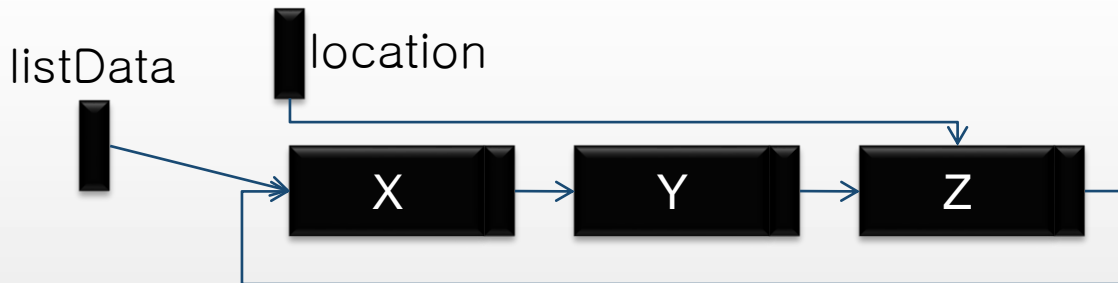
1-help slide

Solution.1



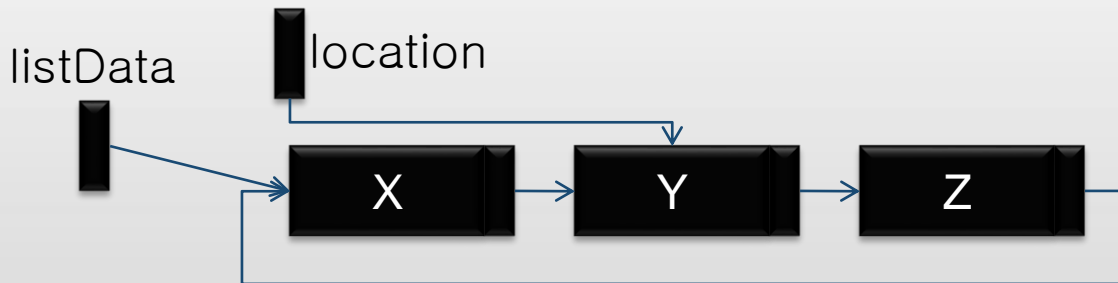
초기상태
location
= listData;

Z 출력



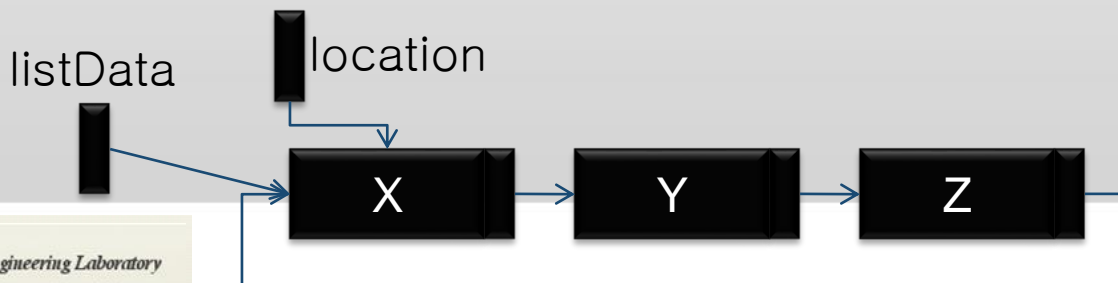
location
는 length-1
만큼 움직입니다.
Z 출력.

Y 출력



location
는 length-2
만큼 움직입니다.
Y 출력.

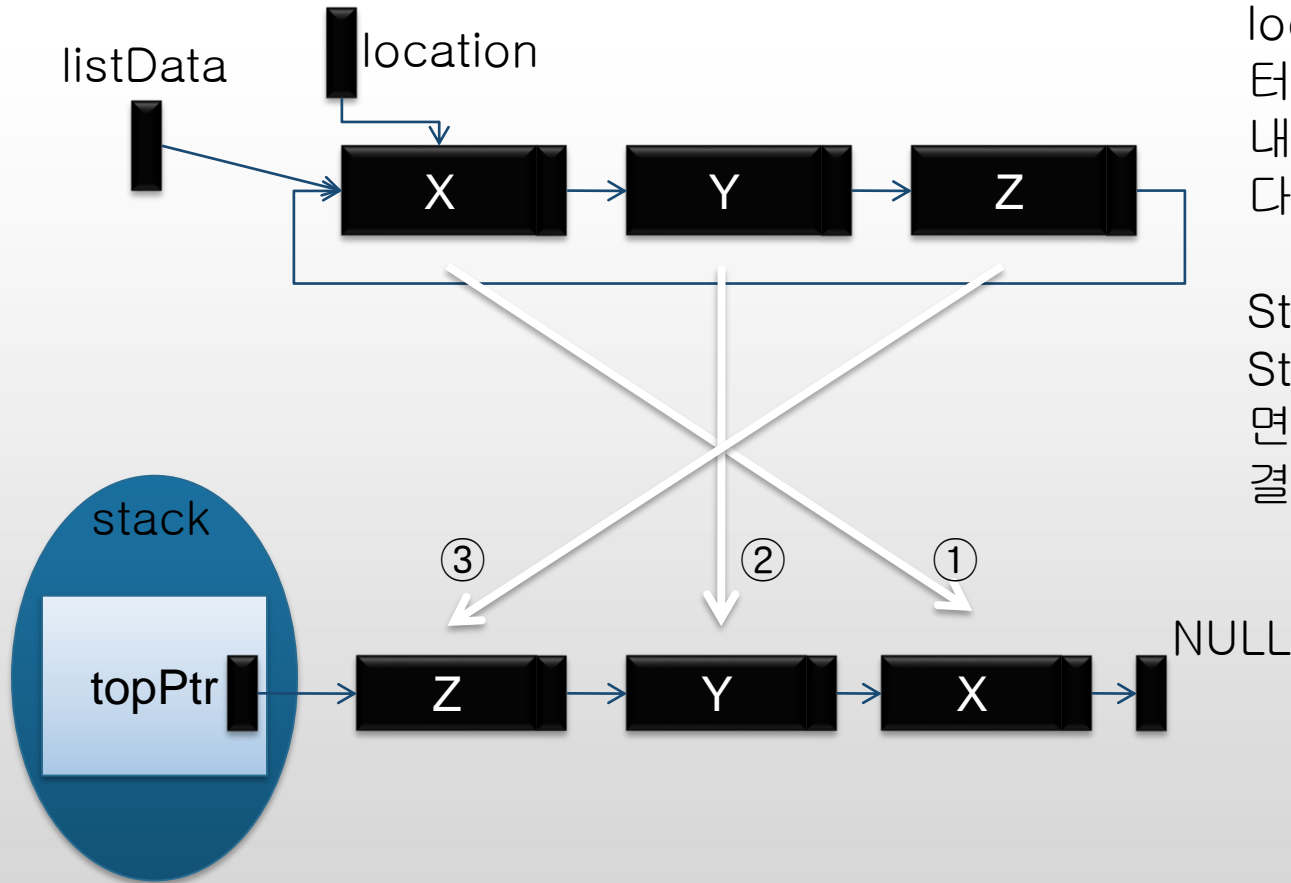
X 출력



location
는 length-3
만큼 움직입니다.
X 출력.

1-help slide

Solution.2



location이 하나씩 데이터를 읽어나가면서 그 내용을 stack에 저장한다.

Stack은 LIFO이므로, Stack을 통해 출력을 하면 입력의 역순으로 된 결과를 얻을 수 있다.

1-help slide

※4번을 마친 뒤 프로그램을 실행하고 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)



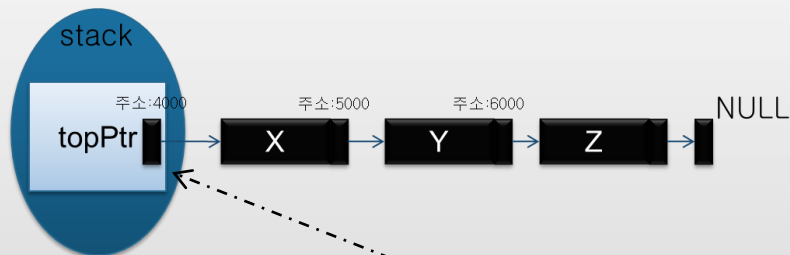
int length = LengthIs();
for(int i=0; i<length; i++)

2. Exercise 9

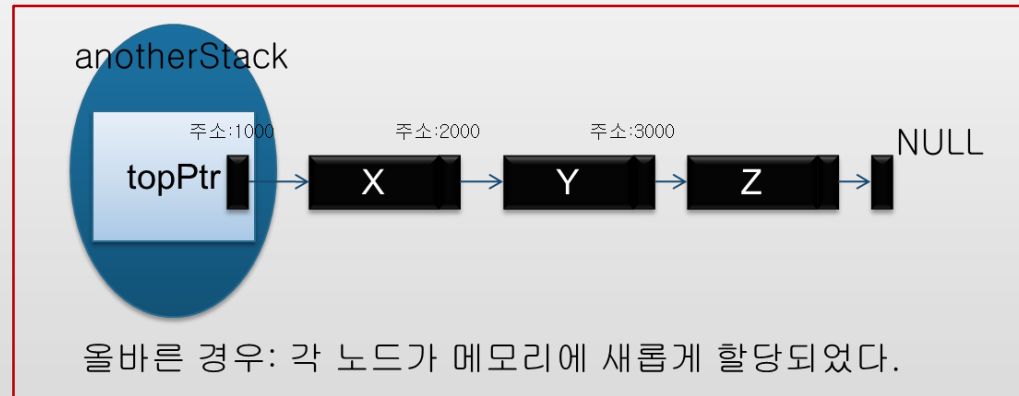
■ Linked 기반으로 구현된 Stack내의 모든 원소를 새로운 스택에 복사하는 Copy함수를 구현하시오.

- ❖ Shallow Copy가 아닌 Deep Copy로 원소를 복사해야함
- ❖ 프로토타입 : `void StackType::Copy(StackType& anotherStack)`

■ 예제



`stack.copy(anotherStack);`
수행 시 Deep copy가 되어야 한다.

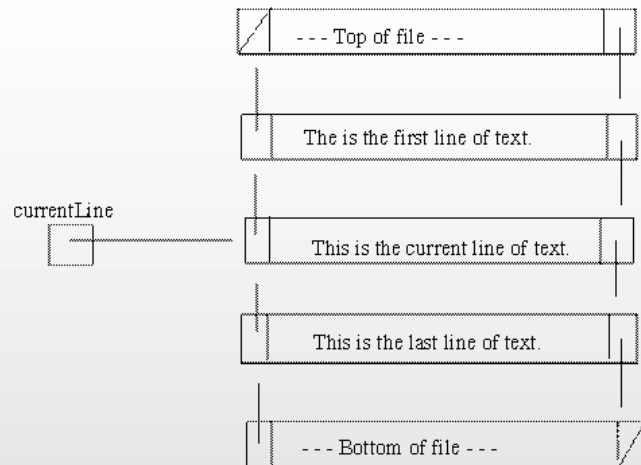


잘못된 경우

3. Exercise 11 (1/2)

- Double Linked List를 사용하여 다음과 같은 구조를 가지는 텍스트 편집기를 구현하시오.

❖ 노드 한 개에서 문자열의 최대 길이는 80으로 제한

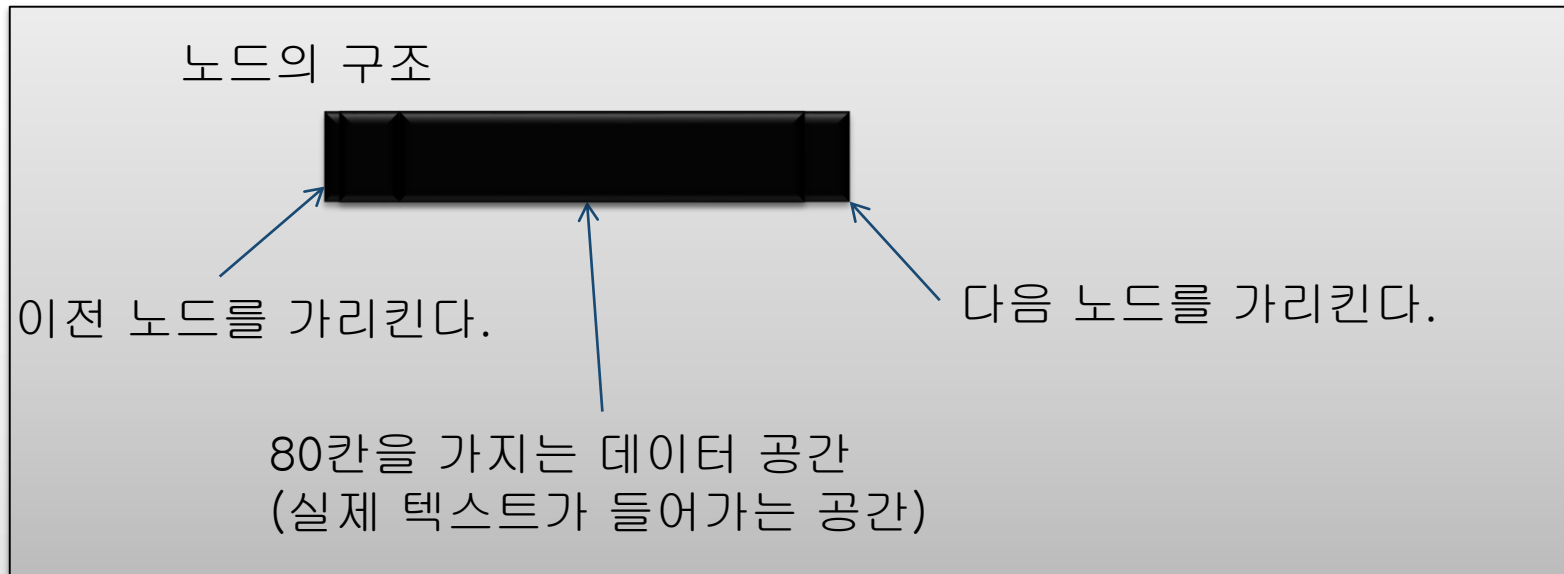


- ❖ a. 위의 구조에 대한 노드 타입을 정의하시오.
- ❖ b. 객체를 초기화 하는 클래스 생성자를 구현하세요

3. Exercise11 (2/2)

- ❖ c. GoToTop함수와 GoToBottom를 작성하시오.
 - GoToTop()
 - Function : 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 first 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
 - Postconditions : 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. 초기화를 하는 클래스 생성자의 구조를 정의 하세요.

```

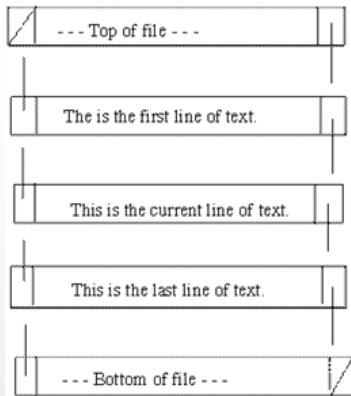
  TextEditor::TextEditor()
  {
    •Header 정의
    •Trailer 정의
    •위 두 노드를 주어진 문장으로 초기화
    •두 노드의 back과 next에 알맞은 값 대입
    •LineType *currentLine이 header를 가리킨다.
    ...
  }

```

생성자는 위와 같은 내용을 가지고 있어야 합니다.

■ C. GoToTop 함수와 GoToBottom를 작성하시오.

❖ GoToTop 수행후



← CurrentLine

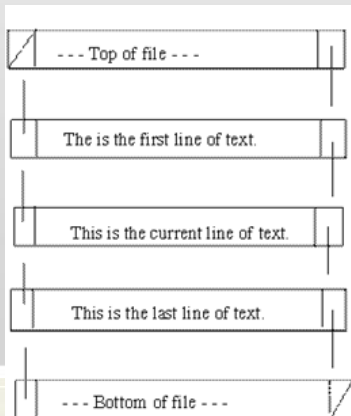
* line이 하나도 없는 경우 trailer를 가리킴)

처음과 끝으로 currentLine을 이동시킬 때 while()문을 이용하세요.

예시

```
GoToTop( )
{
    while(currentLine의 뒤에 노드가 NULL값이 아닌 경우)
        currentLine에 뒤쪽 노드의 값을 대입한다.
    ... //나머지 구현
}
```

❖ GoToBottom() 수행후



← CurrentLine

* line이 하나도 없는 경우 header를 가리킴)

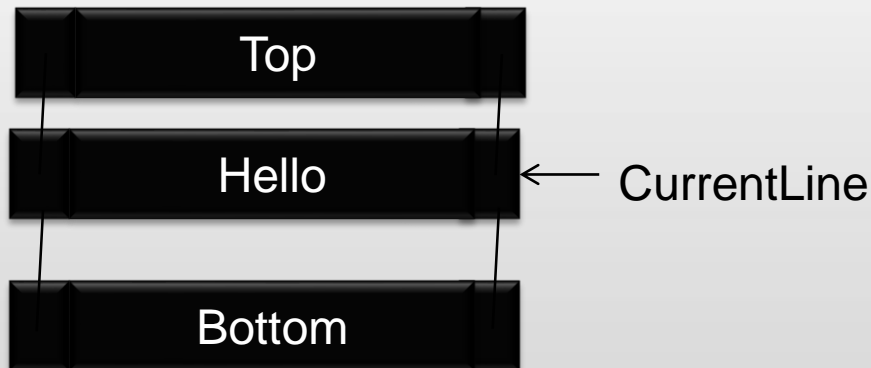
■ d. c의 Big-O를 생각해보고, 위 함수가 $O(1)$ 이 되는 경우를 생각하세요.

- ❖ 구조에 header와 trailer가 있다는 것과 노드가 앞,뒤를 가리킨다는 것을 참고하세요.

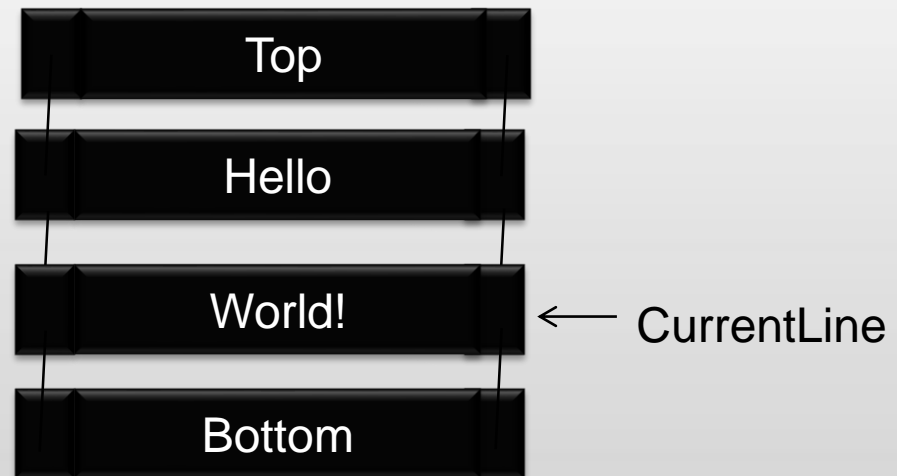
header의 다음 노드가 첫 노드입니다.
trailer의 전 노드가 끝 노드 입니다.

■ E. InsertItem함수를 작성하시오.

예제)



InsertItem 호출 후



현재 CurrentLine 다음에 새로운 노드를 삽입하고 CurrentLine포인터를 이동시킨다. * currentPtr이 trailer를 가리키는 경우에는 trailer 앞쪽에 삽입 (즉 header와 trailer 사이)