

## Lab 03 (3/17) Linked List

- arranged by TA Sangheon Yang

**Due : ~2021.03.17(Thu). 23:59, Late Submission : ~2021.03.18.(Fri) 23:59**

- **Linked List implementation : Insert, Delete, Find, PrintList functions.**
- 아래의 사진과 같이 **[input].txt** 파일을 입력받아 **Linked List**를 생성하고 편집하는 기능을 구현하고, 결과 값을 **[output].txt** 파일에 출력하여 저장한다.
- **Implement the function of creating and editing a linked list by receiving the [input].txt file as input as below and the output result is stored in the [output].txt file.**

```

input1.txt          output1.txt
i 3 -1            insertion(5) failed : can not find the location to be inserted
i 4 3             deletion(9) failed : element 9 is not in the list
i 7 -1            finding(3) failed : element 3 is not in the list
i 5 8             key of the previous node of 7 is head
d 3               key of the previous node of 2 is 7
i 2 7             key:7 key:2 key:4
d 9
f 3
f 7
f 2
p

일반 텍스트 ▾ 템 너비: 4 ▾ 11행, 2열           일반 텍스트 ▾ 템 너비: 4 ▾ 6행, 19열

```

### <Insert>

- 위의 사진에서처럼 "i [insert할 데이터] [insert 할 위치]"로 표현합니다.
- "i 2 7"의 경우 "2"노드를 "7"노드의 NEXT에 위치시키시면 됩니다.
- "i 3 -1"처럼 [insert 할 위치]가 "-1"인 경우, List의 Head의 Next에 위치시켜주시면 됩니다.
- [insert 할 위치]가 List 안에 존재하지 않는 경우, 에러 메시지를 출력해주셔야 합니다.
- Insertion 에러 발생시, 에러메시지 형식은 위 사진과 같이 "insertion([key값]) failed : can not find the location to be inserted"로 출력하셔야 합니다.
- "....".
- In the case of "i 2 7", node "2" should be located at the NEXT of the node "7".
- In the case of location to be inserted is "-1" as "i 3 -1", node "3" should be located at the NEXT of HEAD of the LIST.
- If [location to be inserted] does not exist in the LIST, an error message should be printed out.
- When insertion-error occurred, the format of the error messages should be "insertion([key value]) failed : can not find the location to be inserted" as above.

### <Delete>

- 위의 사진에서처럼 "d [delete할 데이터]"로 표현합니다.
- "d 3"의 경우, List 안에 존재하는 "3"노드를 제거하시면 됩니다.
- 만약 위 사진에서 "d 9"처럼 List 안에 존재하지 않는 노드를 제거하려 할 경우, 에러 메시지를 출력해주셔야 합니다.
- Deletion 에러 발생시 에러메시지 형식은 위 사진과 같이 "deletion([key값]) failed : element [key값] is not in the list"로 출력하셔야 합니다.
- "d [data to be deleted]".
- In the case of "d 3", node "3" should be deleted from the LIST.
- In the case of deleting a node which does not exist in the LIST as "d 9" in the example above, an error message should be printed out.
- When deletion-error occurred, the format of the error messages should be "deletion([key value]) failed : element [key value] is not in the list" as above.

### <Find>

- 위의 사진처럼 "f [search할 데이터]"로 표현합니다.
- "f 2"의 경우, List 안에서 "2"노드의 직전위치의 노드(Parent NODE) 정보를 함께 출력해주셔야 합니다. ex) "key of previous node of [Key값] is [Key값]", "key of previous node of [Key값] is head".
- 만약 위 사진에서 "f 3"처럼 List 안에 존재하지 않는 노드를 search 하는 경우, 에러 메시지를 출력해주셔야 합니다.
- Find 에러 발생시 에러메시지의 형식은 위 사진과 같이 "finding([key값]) failed : element [key값] is not in the list"로 출력하셔야 합니다.
- "f [data to be searched]".
- In the case of "f 2", the previous node(Parent Node) information of node "2" should be printed out as follows : "key of previous node of [Key value] is [Key value]", "key of previous node of [Key value] is head".
- In the case of searching for a node which does not exist in the LIST as "f 3" in the example above, an error message should be printed out.
- When a finding-error occurred, the format of the error messages should be "finding([key value]) failed : element [key value] is not in the list" as above.

### <PrintList>

- 위 사진처럼 "p"로 표현합니다.
- "p"의 경우, List 전체를 출력해주시면 됩니다. 출력시 각 Key당 간격은 [띄어쓰기]입니다.
- 출력순서는 List의 Head부터 순서대로 출력해주시면 되고, List가 비어있을 경우 "empty list!"를 출력해주시면 됩니다.
- "p"

### <Structure & Function Format>

#### • Structure

```
typedef struct Node *PtrToNode;
typedef PtrToNode List;
typedef PtrToNode Position;
typedef int ElementType;
struct Node
{
    ElementType     element;
    Position       next;
};
```

#### • Function

```
List MakeEmpty( List L );
int IsEmpty( List L );
int IsLast( Position P, List L );
void DeleteList ( List L );

void Insert( ElementType X, List L, Position P );
void PrintList(List L);
void Delete( ElementType X, List L );
Position Find ( ElementType X, List L );
```

- 위 사진과 같은 Struct 구조체를 사용하셔야 합니다.
- 위 사진과 같은 함수들을 형식에 맞게 구현해주시면 됩니다.
- **Struct format above should be used for implementation**
- **Functions should be implemented in appropriate format as above.**

### <File Name Format>

- [StudentID].c      ex) 20XXXXXXXXXX.c

### <Execution>

- gcc 20XXXXXXXXXX.c -o 20XXXXXXXXXX
- ./20XXXXXXXXXX [input\_file\_name] [output\_file\_name]
- !!! 꼭 제공되는 testcase로 실행시켜보시기 바랍니다!!!!,

- !!! Run your solution code with the provided test case above and check whether it works properly !!!

#### <Issue>

- 코드 작성 시 꼭 상세하게 주석을 적어주시기 바랍니다. 주석이 없는 경우 Cheating으로 간주될 수 있습니다.
- 에러메시지는 꼭 형식에 맞게 출력해주셔야 합니다. 형식이 맞지 않을 경우 채점시 0점이 될 수 있습니다. 위 예시 사진과 같이 각각의 에러메시지마다 줄바꿈을 해주셔야 합니다.
- 모든 출력 메시지의 알파벳은 소문자만 사용하여 출력 합니다. 줄바꿈을 제외한 모든 공백은 띄어쓰기 한칸입니다.
- 제공된 testcase는 채점 case에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
- 파일 입출력시 파일 이름은 변동될 예정이라, 고정된 파일명을 사용하지 마시고, `argv[]`를 사용하여 구현해주시기 바랍니다.
- 마감 시간 이전의 가장 최신 버전의 commit을 기준으로 채점할 예정입니다.
- 제출 파일과, 폴더 naming 은 꼭 지정된 형식으로 해주셔야 합니다.
- Please write down the detailed comments when writing the code. If there is no comment, it might be considered cheating.
- Error messages must be printed out according to the appropriate format. If not, it can be scored 0. As shown in the example above, you have to change the line for each error message.
- Only lowercase letters should be used for the alphabets in output messages. All spaces except for line change are one space.
- The provided test case is included in the test case for grading. Please check to see if it makes a proper result.
- Do not use a fixed file name when inputting and outputting files, but implement it using `argv[]` as in skeleton code.
- Scoring will be based on the latest version of commit before the deadline.
- The names of the .c file and directory should be named in proper format.

#### <Directory Format>

- 아래와 같이 git 프로젝트 폴더에 "lab03" 폴더 생성후, "lab03" 폴더 안에 "20XXXXXXXXXX.c" 파일을 위치시키시면 됩니다.
- After creating the "lab03" directory in the git-project-directory as below, place the "20XXXXXXXXXX.c" file in the "lab03" directory.

>>>>>>>>>>>>>>>>>>>>>>>>>>>>

2022\_CSE2010\_20XXXXXXXX/ (GitLab project directory)

---lab03/

-----20XXXXXXXXXX.c

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

**QnA : 2021datastructure@gmail.com**