

Lab 02 (3/16) Linked List

- arranged by TA Beomsoo Kim

Due : ~2023.03.22(Wed). 23:59, Late Submission : ~2023.03.23(Thu). 23:59

- **Linked List implementation : Insert, Delete, Find, GetElements.**
- 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
i 5 -1
i 3 5
i 6 3
p
i 6 3
i 1 2
i 2 -1
p
f 2
f 5
f 4
f 3
f 6
p
d 1
d 2
d 3
p

output1.txt
key:5 key:3 key:6
insertion(6) failed : the key already exists in the list
insertion(1) failed : can not find the location to be inserted
key:2 key:5 key:3 key:6
key of the previous node of 2 is head
key of the previous node of 5 is 2
finding(4) failed : element 4 is not in the list
key of the previous node of 3 is 5
key of the previous node of 6 is 3
key:2 key:5 key:3 key:6
deletion(1) failed : element 1 is not in the list
key:5 key:6
    
```

<input> :

- 각 line 마다 Command 가 주어짐.(**insert, delete, find, print**)
- key값은 양의정수. 예외적으로 insert의 경우, 첫번째 **key**값은 양의 정수, 두번째 **key**값은 양의 정수와 -1 (List의 Head를 의미)
- **Commands are given line by line (insert, delete, find, print)**
- **Key values are positive integers. Exceptionally, in the case of insert, the first key value is a positive integer, the second key value is a positive integer and -1 (Head of the List)**

<output> :

각 Command 에 맞는 result 출력.

모든 파일출력은 main 함수에서만 실행

Appropriate output messages for each command.

All file outputs are performed only in the main function.

<MakeEmpty>

- 새로운 header를 만드는 함수입니다.
- List의 포인터를 반환합니다.
- header가 가지는 element의 값은 -1이 되어야 합니다.
- **Make new header.**
- **Return the pointer of List.**
- **The element of the header should be -1.**

<Insert>

- 위의 사진에서처럼 "i [insert할 데이터] [insert 할 위치]" 로 표현합니다.
- "i 3 5" 의 경우 "3"노드를 "5"노드의 NEXT 에 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 않거나 이미 있는 경우, main에 있는 에러 메시지를 출력해주셔야 합니다.
- find() 함수로 [insert할 데이터]를 찾을 때, null값이 아닐 경우는 key중복 에러 메시지를 출력합니다. 그리고 find() 함수로 [insert 할 위치]를 찾을 때, null값일 경우 찾는 key값이 존재하지 않는다는 에러 메시지를 출력합니다.
- key 중복시 에러메시지 형식은 "insertion Key 값 failed : the key already exists in the list" 입니다.
- 찾는 key값이 존재하지 않을 때 에러 메시지 형식은 "insertion key값 failed : can not find the location to be inserted"
- As in the picture above, it is represented as "i [data to be inserted] [location to be inserted]".
- In the case of "i 3 5", place the "3" node in the NEXT of the "5" node.
- If [location to insert] is "-1" like "i 3 -1", place it next to the Head of the List.
- If [location to insert] does not exist or already exists in the List, an error message in main should be printed out.
- When finding [data to insert] with the find() function, if it is not a null value, a duplicate key error message is printed out. And when finding the [location to insert] with the find() function, if it is a null value, an error message is printed out saying that the key value to be searched for does not exist.
- When a key is duplicated, the error message format should be "insertion key value failed : the key already exists in the list"
- When the key value to be searched does not exist, the error message format should be "insertion key value failed : can not find the location to be inserted"

<Delete>

- 위의 사진에서처럼 "d [delete할 데이터]" 로 표현합니다.
- "d 3" 의 경우, List 안에 존재하는 "3"노드를 제거하시면 됩니다.
- 만약 위 사진에서 "d 1" 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 key 값 is not in the list" 로 출력하셔야 합니다.
- As in the picture above, it is represented as "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 1" 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.

<FindPrevious>

- 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 the previous node of key값 is key값", "key of the previous node of key값 is head".
- FindPrevious함수는 List 안에서 찾는 element를 가진 노드의 이전 노드의 Position 포인터를 반환합니다.
- 만약 위 사진에서 "f 4" 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.
- As in the picture above, it is represented as "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 the previous node of Key value is Key value", "key of the previous node of Key value is head".
- The FindPrevious function returns the Position pointer of the previous node of the node that has the element to be found.
- In the case of searching for a node which does not exist in the LIST as "f 4" in the example above, an error message should be printed out.->element->element->element
- 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.

<Find>

- Find함수는 찾는 element를 가지는 노드의 Position 포인터를 반환합니다.
- Find함수는 main함수에서 insert할 element가 이미 존재하는지, insert할 Position을 찾을 수 있는지와 delete할 element가 List안에 있는지 찾기 위해서 사용됩니다.
- The Find function returns the Position pointer of the node that has the element to be found.
- The Find function is used in the main function to find out whether the element to be inserted already exists, whether the Position to be inserted can be found, and whether the element to be deleted is in the List.

<GetElements>

- 위 사진처럼 "p" 로 표현 합니다.
- "p"의 경우, List 전체를 출력해주시면 됩니다. 출력시 각 Key당 간격은 한칸입니다.
- GetElements함수는 모든 리스트의 element들을 main함수 안에서 출력하기 위한 array를 반환합니다.
- GetElements함수는 ListSize 크기의 array를 할당하여 list내부의 element를 앞에서부터 순서대로 복사한 후 그 array를 반환합니다.
- main 함수에서 GetElements함수로 얻은 array에 저장된 값들을 출력하면 됩니다.
- 출력후 array는 메모리 해제를 수행하기 때문에, List에 저장된 element의 포인터를 반환하면 안됩니다.
- 출력순서는 List 의 Head의 Next부터 순서대로 출력해주시면 되고, List가 비어있을 경우 "empty list!" 를 출력해주시면 됩니다.
- As in the picture above, it is represented as "p"
- In the case of "p", you can output the entire list. When printing out, the spacing for

each key is one.

- **The GetElements** function returns **an array** to print all the elements of the list in the main function
- **The GetElements** function allocates an array of the size of ListSize, copies the elements in the list in order from the front, and returns the array.
- Print out the values stored in the array obtained by the GetElements function in the main function.
- **Do not return pointers to elements stored in List as arrays, the array should be deallocated from memory.**
- The order of output is in order from the **next of the Head** of the List, and if the List is empty, print out **“empty list!”**

<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 Insert( ElementType X, List L, Position P );
int* GetElements(List L);
void Delete( ElementType X, List L );
Position Find(ElementType X, List L );
Position FindPrevious ( ElementType X, List L );
void DeleteList ( 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) 20XXXXXXXX.c

<Execution>

- gcc 20XXXXXXXX.c -o 20XXXXXXXX
- ./20XXXXXXXX [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 작성시 주석을 적어주시기 바랍니다. 주석이 없는경우 Cheating으로 간주될 수 있습니다.

- 제공된 **testcase**는 채점 **case**에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
- 파일 입출력은 **argv[]**를 사용하여 구현해주시기 바랍니다.
- 제출 마감 시간 이전의 가장 최신 버전의 **commit**을 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.
- 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.
- 출력시, 위 사진의 예시와 같은 형식으로 출력해주시면 됩니다. 모든 공백은 띄어쓰기 한칸입니다. 모든 출력 메시지의 알파벳은 소문자만 사용하여 출력합니다.
- -> 출력시 위 사진과 같이 커맨드에 알맞는 메시지 출력후 줄바꿈.
- All the messages must be printed out according to the appropriate format as shown in the example above. All spaces are one space. Only lowercase letters should be used for the alphabets in output messages.
- -> newline(\n) after each output message.

<Directory Format>

- 아래와 같이 git 프로젝트 폴더에 "lab02" 폴더 생성후, "lab02" 폴더 안에 "20XXXXXXXXX.c" 파일을 위치시키시면 됩니다.
- After creating the "lab02" directory in the git-project-directory as below, place the "20XXXXXXXXXXXX.c" file in the "lab02" directory.

[illegible]

2023 CSE2010 20XXXXXXXX/ (GitLab project directory)

---lab02/

```
-----20XXXXXXXXXX.c
```

[illegible]