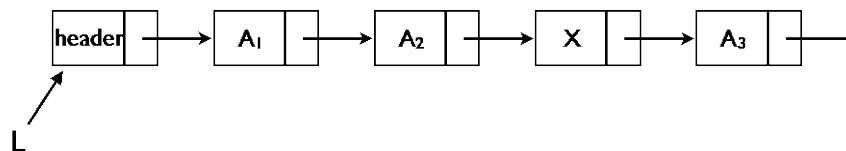


Lab 3: Linked List

Implement List ADT in this lab session. As we discussed in the class, the linked list is a list of nodes that are linked as shown below. Your List ADT has four main operations such as 1) insert, 2) delete, 3) find the previous, and 4) show the list. In addition, your List ADT needs to have functions of MakeEmpty, IsEmpty, IsLast, Find, and DeleteList.

- **Insert** a new node right after the node with the given key. If your list does not have any node with the given key, just print an error message.
- **Delete** the node with the given key. If your list does not have any node with the given key, just print an error message.
- **Find the previous node** of the node with the given key. If your list does not have any node with the given key, just print an error message.
- **Show the entire list.** If your list is empty, just print that your list is empty.



1. Input

Obtain a list of operations from the given input file, and execute the given operations in order. A detailed specification of the operations is provided below. Each line represents a single operation. Each operation and the necessary parameters are separated by a space. You may assume that there are no duplicate keys in the list and the input. You may also assume that the input key values (represented as x and y below) are any positive integers.

- **i x y**: insert a new node with the key “x” after the node with the key “y”
- **i x -1**: insert a new node with the key “x” before the first node in the list
- **d x**: delete the node with the key “x”
- **f x**: print the key of the previous node of the node with the key “x”
- **p**: print the entire list from the beginning to the end

An input file and the corresponding result are shown below.

```
thkim@ubuntu:/data/thkim/dslab$ ./p2 lab2_input.txt
Insertion(5) Failed : element 8 is not in the list
Deletion failed : element 9 is not in the list
Could not find 3 in the list
Key of the previous node of 7 is header.
Key of the previous node of 2 is 7.
key:7 key:2 key:4
```

2. List ADT

(1) Data Specification for the objects

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

(2) Function specification

- List MakeEmpty(List L);
- int IsEmpty(List L);
- int IsLast(Position P, List L);
- void Delete(ElementType X, List L);
- Position FindPrevious (ElementType X, List L);
- Position Find(ElementType X, List);
- void Insert (ElementType X, List L, Position P);
- void DeleteList (List L);

3. Program description

- name : p3.c
- input: a list of operations in a file (an input file name is given as a command line argument. See an example in “1. input” on the first page)
- output : the corresponding result on the standard output

Submit to the course website (<https://portal.hanyang.ac.kr>) your source code and a written report. Your report should include the description of your own implementation.