HW2: Linked List

제출은 gitLab을 통해서 하며 **http://hconnect.hanyang.ac.kr/2017\_CSE2010\_수업번호/2017\_CSE2010\_수업번호\_학번.git 에 HW2라는 폴더를 만들어 진행.**

\* 프로그램 제출간 유의사항

- 소스코드에는 주석이 있어야 함.

- 주어진 구조체와 input.txt를 사용해야 함

**- 숙제의 소스코드 평가는 linux ubuntu 16.04.2 LTS 버전 gcc 5.4.0에서 함.**

\* 보고서 제출간 유의사항

- 작성한 소스 코드가 첨부되어야 하며, 실행결과가 첨부되어야 함.

- 분량은 제한이 없으나 1~2apge로 간략하게 설명.

- 보고서는 hw2\_학번.확장자(doc, docx, pdf)로 제출.

**제출시간: '17.3.29(23:59) 까지**

**\*지연제출**

- 24시간 이내는 해당 과제 50% 감점, 48시간 이내는 75% 감점.

- 지연제출자는 E-mail(casualab@hanyang.ac.kr)과 gitlab에 모두 제출.

- E-mail제목: "hw2\_학번\_자신의 수업 요일(수, 목)\_이름 " 형식으로 제출.

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, 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.

|  |
| --- |
| Macintosh HD:private:var:folders:n2:r1pzhmbn34vg4rtp4fbzwg0w0000gn:T:Screenshot_3_8_15_10_46_PM.jpg |

1. List ADT

(1) Data Specification for the objects

typedef struct Node\* PtrToNode;

typedef PtrToNode List;

typedef PtrToNode Position;

typedef struct ElementType{

int key;

float value;

}ElementType;

struct Node{

ElementType element;

Position next;

};

3. Program description

* name : hw2\_학번.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
* Fill in the blank given source code.
  1. void PrintList(List L);

void Delete(Element X, List L);

* 1. void Insert(ElementType X, List L, Position P){
  2. value setting part
  3. }
  4. int main() { case 'i': ~}

Submit to the course gitlab website (http://hconnetion.hayang.ac.kr) your source code and a written report. Your report should include the description of your own implementation.