

## Homework#4 Doubly Linked List Exercise (due: 4 월 27 일)

### ● Requirement:

1) Menu 로 구성할 것

(1.insert, 2.delete, 3.backward, 4.forward, 5.search, 6.quit)

2) Input data 는 숫자 및 이름이며, 숫자중심 오름차순으로 입력됨

### 1. Node 구성:

Struct info { //C	class Node { //C++
int data;	<b>int data; string name;</b>
<b>char name[10];</b>	Node *next; Node *prev;
struct info *next;	Node(int val, string str) {
struct info *prev;	data=val; name=str; next=0; prev=0;}
};	friend class List; }

### 2. ADT

1) insert(): case 1) if (head==null) create head node  
2) else if (temp->data < head->data) insert front of head node  
3) else { /\* find place to insert  
p = head; q=head;  
while ((p != NULL) && (p->data < temp->data)) {  
q=p; p=p->next;}  
if (p!= NULL) insert in the middle  
else insert last}

### 2) Delete():

“ Enter number to delete: (num)”  
case1: if (head==null) “List empty”  
2: else if (head->data == num)  
if (head->next !=null) move head and delete  
else delete head node  
3: else{ /\* insert 와 같은 방법으로 이동한후  
if (p->data == num)  
if (p->next != null) delete node & link  
else /\* delete last node  
4: else “Not found”

- 3) backward():   if (head ==NULL)   “List is empty”  
                   else /\* 마지막 노드로 이동한후  
                           마지막 노드부터 처음 노드까지 이동하면서  
                           전체리스트를 출력할것   -- Backward Listing ---
- 4) forward():    if (head ==NULL)   “list empty”  
                   else   처음부터 끝까지 전체 리스트 출력할것 -- Forward listing --
- 5) search():      “Enter number :   (num)”  
                   리스트의 처음부터 끝까지 scan 하면서  
                   입력데이터 X 를 찾으면 “X is found”  
                   아니면   “ X is not found” 를 출력할것
- 6) Quit():   Terminate all created node and   exit program

### 3. 테스트

- 1) Insert →      10 Kim
- 2) Insert →      20 Lee
- 3) Forward→    10 Kim,   20 Lee
- 4) Backward →   20 Lee,   10 Kim
- 5) Search 10 →   10 is in the List
- 6) Search 30 →   30 is not in the List
- 7) Delete 10 →   10 is deleted
- 8) Delete 20 →   20 is deleted
- 9) Forward →    Empty List