

Lab#5 Singly Linked List 연습

Main Program:

1. 메뉴 (1. insert, 2. delete, 3. list, 4. search, 5. PrintLast
6. PrintNth 7. InvertList 8. Quit)
2. Quit 선택 할 때까지 계속할 것
3. Node 의 구성 (2 개의 filed; data, link)

Singly Linked-List ADT

- 1) Insert: 새로운 숫자 입력시, 숫자값에 따라 오름차순으로 list에 삽입 할 것
 - (1) head == NULL => 새로운 노드 생성
 - (2) (head != NULL & new number < head->num) => head 이동
 - (3) (head != NULL & new number > head->num) => 전체리스트 검색 후 적합한 장소에 삽입한다.
- 2) Delete: delete 할 노드의 number 값으로 해당 노드를 찾아서 삭제할것
 - (1) head == NULL=> "리스트 empty" 출력
 - (2) (head != NULL & (number == head->num))=> head 이동
 - (3) (head != NULL & (number != head->num))=> 전체리스트 검색후 해당노드를 삭제한다. 못 찾을 경우는 "not found" 출력.
- 3) List: head 노드가 NULL 이면, print "List empty"
아니면, 리스트의 전체 노드들을 출력 (p->num)
- 4) Search: 찾고싶은 노드의 숫자를 입력 받는다.
해당 숫자의 리스트에 존재여부를 출력한다.
(예: 10 is in the list 또는 10 is not in the list)
- 5) PrintLast: Head 가 NULL 이면 print "List Empty"
아니면, 리스트의 마지막 노드만 출력한다.
- 6) PrintNth: Head 가 NULL 이면 print "List Empty"
아니면, 리스트의 N 번째 노드만 출력한다.

7) InvertList: Head 가 NULL 이면 print “List Empty”

아니면, 리스트의 마지막노드부터 처음노드 까지 출력한다

8) Quit: 리스트의 모든 노드들을 free 시켜야 한다.

<<TEST 예: >>

입력:

Delete 10

-> insert 10 -> insert 5 -> list

-> insert 30 -> search 5

-> delete 5 -> list

-> Find3rd ->

-> PrintLast ->

-> insert 40 ->invertList

-> quit;

출력:

not found

list: 5 10

5 is in the list

list: 10 30

not found

30

List: 40 30 10