

# CSE2010 자료구조론

## Week 3: Linked List

---

한양대학교 ERICA  
소프트웨어융합대학  
ICT 융합학부



# 오늘의 실습

---

- Linked List 직접 구현하기
- 총 6개의 함수
  - insert\_node(ListNode \*\*phead, ListNode \*p, ListNode \*new\_node)
  - insert\_node\_after\_value(ListNode \*head, int value, ListNode \*new\_node)
  - remove\_node\_by\_value(ListNode \*\*phead, ListNode \*head, int value)
  - remove\_node(ListNode \*\*phead, ListNode \*p, ListNode \*removed)
  - display(ListNode \*head)
  - ListNode \*concat(ListNode \*head1, ListNode \*head2)

# 함수 설명

---

- `insert_node_after_value(ListNode *head, int value, ListNode *new_node)`
  - 특정한 값을 뒤에 새로운 노드를 이어붙이기
- `remove_node_by_value(ListNode **phead, ListNode *head, int value)`
  - 특정한 값을 가진 노드를 삭제하기
  - 단 가장 앞에 있는 노드와 가장 끝에 있는 노드를 지웠을 경우도 생각하기
- 나머지는 강의자료 참조...

```
#include <stdio.h>
#include <stdlib.h>

typedef int element;
typedef struct ListNode {

} ListNode;

void insert_node(ListNode **phead, ListNode *p, ListNode *new_node) {

}

void insert_node_after_value(ListNode *head, int value, ListNode *new_node) {

}

void remove_node_by_value(ListNode **phead, ListNode *head, int value) {

}

void remove_node(ListNode **phead, ListNode *p, ListNode *removed) {

}

void display(ListNode *head) {

}

ListNode *concat(ListNode *head1, ListNode *head2) {

}
```

```
int main(int argc, const char * argv[]) {

    struct ListNode *head1 = malloc(sizeof(struct ListNode));
    struct ListNode *head2 = malloc(sizeof(struct ListNode));
    struct ListNode *head3 = malloc(sizeof(struct ListNode));

    head1 = NULL;
    head2 = NULL;
    head3 = NULL;

    for (int i=1;i<=5;i++) {
        struct ListNode *new_node = malloc(sizeof(struct ListNode));
        new_node->link = NULL;
        new_node->data = i;
        insert_node(&head1, NULL, new_node);
    }
    for (int i=6;i<=10;i++) {
        struct ListNode *new_node = malloc(sizeof(struct ListNode));
        new_node->link = NULL;
        new_node->data = i;
        insert_node(&head2, NULL, new_node);
    }

    printf("링크드리스트1 전체출력\n");
    display(head1);
    printf("링크드리스트2 전체출력\n");
    display(head2);
    printf("링크드리스트 합치기\n");
    head3 = concat(head2, head1);
    display(head3);

    printf("새로운 노드 11을 4 뒤에 넣기\n");

    struct ListNode *new_node = malloc(sizeof(struct ListNode));
    new_node->link = NULL;
    new_node->data = 11;

    insert_node_after_value(head3, 4, new_node);
    display(head3);

    printf("새로운 노드 11을 0 (없는 값) 뒤에 넣기\n");

    struct ListNode *new_node2 = malloc(sizeof(struct ListNode));
    new_node2->link = NULL;
    new_node2->data = 11;

    insert_node_after_value(head3, 0, new_node2);
    display(head3);

    printf("노드 11을 삭제하기\n");
    remove_node_by_value(&head3, head3, 11);
    display(head3);
    printf("노드 1을 삭제하기\n");
    remove_node_by_value(&head3, head3, 1);
    display(head3);
    printf("노드 10을 삭제하기\n");
    remove_node_by_value(&head3, head3, 10);
    display(head3);
    printf("노드 1을 삭제하기\n");
    remove_node_by_value(&head3, head3, 1);
    display(head3);

    return 0;
}
```

# 실행예제

```
링크드리스트1 전체출력
5 4 3 2 1
링크드리스트2 전체출력
10 9 8 7 6
링크드리스트 합치기
10 9 8 7 6 5 4 3 2 1
새로운 노드 11을 4 뒤에 넣기
10 9 8 7 6 5 4 11 3 2 1
새로운 노드 11을 0 (없는 값) 뒤에 넣기
해당 숫자가 없음.
10 9 8 7 6 5 4 11 3 2 1
노드 11을 삭제하기
10 9 8 7 6 5 4 3 2 1
노드 1을 삭제하기
10 9 8 7 6 5 4 3 2
노드 10을 삭제하기
9 8 7 6 5 4 3 2
노드 1을 삭제하기
해당 숫자가 없음.
9 8 7 6 5 4 3 2
Program ended with exit code: 0
```

# 수고하셨습니다.

---