

Data Structure in C
Homework #3
Due 11:59 PM, 5 Dec (Sun), 2021

1. 단순 연결 리스트 (Singly Llnked List)를 이용하여 숫자들을 항상 정렬된 상태로 유지하는 리스트 SortedList 를 구현한다. 기능으로서 아래 함수들을 구현한 후, 간단한 테스트 함수 (main function)를 만들어 테스트 하시오 (단, 정렬은 오름차순을 기준함): [70 pt]

add_node(Node *head, int item) : 정렬된 리스트에 새로운 요소 추가
delete_node(Node *head, int item) : 정렬된 리스트에서 item 제거
clear(Node *head) : 리스트의 모든 아이템 삭제
is_in_list(Node *head, int item) : item이 리스트 안에 있는지를 검사하여, 있으면 TRUE (1)를 리턴하고 없으면 FALSE(0)를 반환
count_list(Node *head) : 리스트 내의 노드 개수 (즉, 아이템 개수)를 반환
is_empty(Node *head) : 리스트가 비었는지를 검사하여, 비었으면 TRUE (1)를 반환하고 아이템이 존재하면 FALSE(0)를 반환
print_list(Node *head) : 리스트의 모든 데이터 값을 모니터에 출력

Note 1) 테스트 함수는 무한 루프를 이용하여 위의 함수들을 메뉴 (예: 1. Add; 2. Delete; 3. Clear; ...) 를 이용해 테스트하는 프로그램을 작성함.

Note 2) 단순 연결 리스트의 노드는 아래와 같은 구조를 이용하며, 구현 용이성을 위해 dummy 노드 가능함. 또한, 필요시 함수명과 패러미터를 변경해도 괜찮음.

```
typedef struct node{
    int data;
    struct node *link;
} Node;
```

2. **Dummy node** 와 **이중 연결 리스트**를 이용하여 최근 실행한 모바일 앱 이름을 저장하는 app history 관리 프로그램을 작성한다. Dummy node로는 Header와 Trailer 두 개의 노드를 만들고, 새로운 노드를 추가할 경우 Header 뒤에 추가한다 (즉, 가장 오래된 앱 이름이 Trailer 바로 앞에 저장되고, 가장 최근에 사용한 앱이 Header 다음의 첫번째 노드로 저장된다). 이중연결리스트의 기능으로서 다음 함수들을 구현한 후, 간단한 테스트 함수 (main function)를 만들어 테스트 하시오 (테스트 함수는 1번 문제와 같이 무한 루프를 이용하여 위의 함수들을 메뉴를 이용해 테스트하는 프로그램을 작성함. 앱 리스트는 최소 5개 이상을 테스트할 것. 예: Kakaotalk, Instagram, Phone, Mail, Google, ...) [70 pt]

add_node(Node *head, char *str) // 리스트에 데이터 추가 (더미 헤드 노드 바로 다음에 추가)
delete_node(Node *head, Node *p) // p 포인터가 가리키는 노드를 리스트에서 삭제
is_in_list(Node *head, char *str) : item이 리스트 안에 있는지를 검사하여, 있으면 TRUE (1)를 리턴하고 없으면 FALSE(0)를 반환
count_list(Node *head) // 리스트내의 노드 개수를 반환 (더미 노드 제외)
is_empty(Node *head) : 리스트가 비었는지를 검사하여, 비었으면 TRUE (1)를 반환하고 아이템이 존재하면 FALSE(0)를 반환
print_list(Node *head) // 리스트내의 노드 데이터들을 헤드 노드 다음부터 출력 (즉, 가장 최근 사용한 앱부터 출력)
search(Node *head, char *str) // 리스트내의 노드 데이터들을 차례대로 검색해서 str 값과 일치하는 노드가 있으면 해당 노드의 주소를 반환; 일치하는 노드가 없을 경우 NULL 반환