

Homework #1: Linked list 개발

● 과제 내용

- Linked list 프로그램을 구현하고 실험한다. 주어진 데이터를 linked list에 모두 저장하고, 새로운 데이터 추가, 기존 데이터 삭제, 검색, 리스트의 데이터 순서를 역순으로 재배치 하는 작업을 실험해본다.
- 삽입할 데이터는 텍스트 파일 students.txt에 들어 있는 학생 명단이며 각 학생의 명단은 이름(문자열)과 학번(정수)로 구성된 구조체로 아래와 같이 정의한다.

```
typedef struct Listdata {
```

```
    char name[30];
```

```
    int id;
```

```
} listData;
```

- 하나의 리스트 노드의 정의는 다음과 같다.

```
typedef struct ListNode {
```

```
    listData data;
```

```
    struct ListNode* link;
```

```
} listNode;
```

- 리스트의 첫 번째 노드를 가리키는 head는 다음과 같이 정의한다.

```
typedef struct LinkedList {
```

```
    listNode* head;
```

```
    int length;
```

```
} linkedList;
```

● 작업 수행 순서

(1) 리스트 저장

- 프로그램을 실행하면 먼저 주어진 scores.txt 파일 안의 모든 점수를 linked list에 순서대로 저장한다. 프로그램 실행시 초기에 파일의 모든 내용을 리스트에 삽입하기 위해 마지막 노드로 데이터를 저장하는 insertLast()를 작성하여 반복적으로 호출하여 사용하도록

한다.

함수 정의: void insertLast(linkedList* L, listData* item)

(2) 테스트

- 명령문을 입력 받아 이를 수행하는 작업을 반복하도록 한다.
- 명령문 입력 시, scanf 를 사용하지 않도록 하며 대신에 gets 로 한 줄을 읽어 온 후 먼저 명령을 파악하고 그에 따라 후속되는 arguments 를 알아내는 방법을 사용하도록 한다.
- 명령문 종류 예시

print 설명: 리스트의 내용을 차례대로 출력한다. 함수 정의: void printList(linkedList* L)
search 202112345 설명: 리스트에서 학번이 202112345인 노드를 찾아 학생의 이름을 출력하고 찾은 노드의 주소를 리턴한다. 탐색에 실패할 경우 탐색 실패 메시지를 출력하고 NULL을 리턴한다. 함수 정의: listNode* search(linkedList* L, int x)
insert 202154321, 202198765, "홍길동" 설명: 리스트에서 search() 함수를 통해 학번이 202154321인 노드를 먼저 찾고, 해당 노드 뒤에 학번이 202198765, 이름이 "홍길동" 인 노드를 추가한 후 리스트의 전체 내용을 차례대로 출력한다. 삽입에 실패할 경우 삽입 실패 메시지를 출력한다. 함수 정의: int insert(linkedList* L, listNode* pre, int x)
delete 202124680 설명: 리스트에서 search() 함수를 통해 학번이 202124680인 노드를 먼저 찾고, 해당 노드 삭제한 후 리스트의 내용을 차례대로 출력한다. 삭제에 실패할 경우 삭제 실패 메시지를 출력한다. 함수 정의: listNode* delete(linkedList* L, int x)
reverse 설명: 리스트의 노드 순서를 역순으로 바꾼 후 리스트의 내용을 차례대로 출력한다. 정렬이 아님을 주의한다. 함수 정의: void reverse(linkedList* L)
getLength 설명: 리스트의 노드 개수를 출력한다. 함수 정의: int getLength(linkedList* L)
exit 설명: 프로그램을 종료한다.

- 제출

- 프로그램 소스 코드와 보고서

- 보고서 포함 내용

- 모든 명령에 대해 수행해 본 실행창을 캡처한 이미지

- 과제 수행 중 어려웠던 점, 좋았던 점을 보고서에 간단하게 기술