

# 과제 1

2020 년 1 학기 자료구조 및 알고리즘

---

## 목표

사용자로부터 명령어를 입력받아 100 개의 리스트에 대한 데이터 삽입/삭제/탐색/출력을 수행하는 프로그램을 작성한다. 주어진 뼈대 코드(main.h 와 main.c)를 활용하되, main.c 에 비어있는 함수를 채워넣는 식으로 구현한다.

## 주의사항

- 모든 명령어는 알파벳 소문자로 주어진다.
- 프로그램 내부에는 최대 100 개의 리스트가 있을 수 있다. 따라서, 아래 명세에서 <id> 값은 0 이상 99 이하의 정수이다.
- 리스트는 4 바이트 정수형 데이터를 저장한다. 따라서, 아래 명세에서 <data> 값은 항상 4 바이트 정수형 자료형으로 입력 및 표현할 수 있다.
- 예외적으로 출력 명세에 언급한 경우가 아니라면 항상 유효한 입력만 주어진다고 가정해도 좋다.
- 아래 예제 명세에서 😊 기호의 왼편은 예제 입력, 오른편은 예제 입력에 대한 예제 출력을 나타낸다.
- 한 명령어에 대한 출력이 끝나면 줄 바꿈 문자를 출력하여 다음 명령어의 입력이 콘솔의 맨 왼쪽 끝에서 이루어질 수 있도록 하라.

## 명세

프로그램은 표준 입력(scanf 를 이용)을 통하여 사용자에게 명령어를 입력 받고, 결과를 표준 출력(printf 를 이용)으로 출력한다. 종료 명령어가 입력될 때까지 반복적으로 명령어를 입력받아 처리해야 한다. 프로그램에서 지원하는 명령어는 아래와 같다.

insert <id> <pos> <data>

---

**동작** <id>번째 리스트의 <pos> 위치에 <data>를 삽입한다. <id>번째 리스트의 길이가 n 일 때 <pos>는 0 이상 n 이하의 정수이며 0 일 경우 데이터를 리스트의 머리 앞에 (새로운 머리가 된다), n 일 경우 데이터를 리스트의 꼬리 뒤에(새로운 꼬리가 된다) 삽입한다.

예외적으로 <pos>의 값이 -1 일 때, 리스트의 꼬리 뒤에 삽입하는 것으로 간주한다.

만약, <id>번째 리스트가 비어있을 경우 유효한 <pos>의 값은 0 또는 -1 뿐이다. 두 경우 모두 <data>는 리스트의 머리이자 꼬리가 된다.

**출력** <pos>의 값이 <id>번째 리스트의 길이보다 크면 삽입에 실패하여 -1 을 출력하고, 삽입에 성공하면 1 을 출력한다.

**예제** insert 0 0 3 😊 1

insert 0 -1 5 😊 1

insert 1 3 2 😊 -1 (리스트 1 은 비어있으므로 3 번째 인덱스에 삽입할 수 없다)

## find <id> <data>

---

**동작** <id>번째 리스트에서 <data>를 찾아 그 인덱스를 출력한다. 만약 <data>가 여러 번 등장한다면 제일 작은 인덱스를 한 번만 출력하면 된다. 출력되는 인덱스는 0 부터 시작한다. 따라서, 리스트의 길이가 n 일 때 머리에서 데이터를 찾으면 0 을, 꼬리에서 데이터를 찾으면 n-1 을 출력해야 한다. 찾지 못할 경우 -1 을 출력한다.

**출력** <id>번째 리스트에서 처음 등장하는 <data>의 인덱스를 출력하고(0 부터 시작), 만약 찾지 못했다면 -1 을 출력한다.

**예제** find 0 5 😊 1

find 0 4 😊 -1 (리스트 0 에는 위의 insert 예제에서 처럼 3 과 5 가 있으므로 4 는 존재하지 않음)

## delete <id> <pos>

---

**동작** <id>번째 리스트에서 <pos>번째 데이터를 삭제한다. <id>번째 리스트의 길이가 n 일 때 <pos>는 0 이상 n 미만의 정수이며 0 일 경우 리스트의 머리를, n-1 일 경우 리스트의 꼬리를 삭제한다.

예외적으로 <pos>의 값이 -1 일 때, 리스트의 꼬리를 삭제하는 것으로 간주한다.

**출력** <pos>의 값이 <id>번째 리스트의 길이보다 크거나 같으면 삭제에 실패하여 -1 을 출력하고 삭제에 성공하면 1 을 출력한다.

**예제** delete 0 0 😊 1 (리스트 0 에는 현재 3 5 가 있는데 이 delete 명령어를 수행하면 5 만 남게 된다)

delete 1 0 😊 -1 (리스트 1 은 비어있으므로 0 번째 요소를 삭제할 수 없다)

## count <id>

---

**동작** <id>번째 리스트의 길이를 출력한다.

**출력** <id>번째 리스트의 길이를 출력한다. 비어있다면 0 을 출력한다.

**예제** count 0 😊 1

## reset <id>

---

**동작** <id>번째 리스트의 데이터를 모두 삭제하고 초기화한다.

**출력** 이 명령어의 출력은 없다.

**예제** reset 0

## print <id>

---

**동작** <id>번째 리스트에 저장된 데이터를 앞에서부터 하나씩 출력한다.

**출력** <id>번째 리스트에 저장된 데이터를 앞에서부터 하나씩 띄어쓰기로 구분하여 출력한다. 구현의 편의를 위해 마지막 데이터를 출력하고 그 다음에 띄어쓰기를 출력하는 것을 허용한다. 만약 리스트가 비어있다면 줄바꿈 문자 하나만 출력하라.

**예제** (0 번째 리스트에 3, 4, 5 가 저장되어 있다고 가정)

print 0 😊 3 4 5

## print\_reverse <id>

---

**동작** <id>번째 리스트에 저장된 데이터를 뒤에서부터 하나씩 출력한다.

**출력** <id>번째 리스트에 저장된 데이터를 뒤에서부터 하나씩 띄어쓰기로 구분하여 출력한다. 구현의 편의를 위해 마지막 데이터를 출력하고 그 다음에 띄어쓰기를 출력하는 것을 허용한다. 만약 리스트가 비어있다면 줄바꿈 문자 하나만 출력하라.

**예제** (0 번째 리스트에 3, 4, 5 가 저장되어 있다고 가정)

print\_reverse 0 😊 5 4 3

## exit

---

**동작** 프로그램을 종료한다.

**출력** 이 명령어의 출력은 없다.

**예제** exit (프로그램 종료)

## 제출

- 기한: 2020 년 5 월 10 일 일요일 23:59
- 방법: 포털의 과제 제출란에 정해진 이름으로 압축 파일을 올린다. 학번과 이름이 20171001 김덕성이라면 **HW1\_20171010\_김덕성.zip** 으로 아래 파일을 압축하여 제출한다.
- **main.c**: 위의 명세를 구현한 소스 코드 파일
- **report.pdf**: 구현 방법을 요약한 보고서. 단, 보고서는 A4 용지로 2 장 이내로 제한한다.
- 딜레이는 전체 점수에서 1 일 이내(5 월 11 일 23:59)인 경우 30%, 3 일 이내(5 월 13 일 23:59)인 경우 50%, 7 일 이내(5 월 17 일 23:59)인 경우 70%를 감점한다.

## FAQ

- 수업시간에 배운 예제 코드를 참고해도 되나요?  
참고해도 됩니다.
- 새로운 파일을 만들거나 main.h 의 내용을 수정해도 되나요?  
안됩니다. 반드시 main.c 안에 허용된 부분만 수정하시기 바랍니다. main.c 안에 새로운 함수를 추가하거나, 전역 변수를 선언하는 것은 허용하지만 그러지 않고도 구현할 수 있습니다.
- 프로그램 내부에 리스트가 여러개 있어야 할 것 같은데, 리스트의 리스트를 만들어야 하나요?  
아닙니다. 각, 리스트는 이중 연결 리스트로 구현하되, 전체 리스트의 개수는 고정되어 있기 때문에(100 개) 뼈대 코드에 있는 것 처럼 이중 연결 리스트 구조체의 배열을 사용하는 것을 권장합니다.
- 명세에 문제가 있는데요?  
포털의 커뮤니티 게시판에 올려주시기 바랍니다.
- 명세에 문제가 있는 것 같은데 소스 코드를 보셔야할 것 같은데요?  
이메일로 코드를 보내주시기 바랍니다.
- 혹시 구현의 정확성을 확인할 수 있는 테스트 입력/출력이 있나요?  
제가 직접 제공해드릴지 말지는 고민 중입니다. 수강생이 직접 만든 테스트 입력/출력을 커뮤니티에 공유하는 것은 허용합니다.
- 다른 수강생과 과제에 대한 내용을 공유해도 되나요?  
논의는 허용하나, 소스 코드의 일부를 공유하는 것은 금지합니다.
- 코드의 시간복잡도를 고려해야 하나요?  
리스트에 임의 접근(random access)이 불가능하기 때문에 이로 인한 탐색 시간은 괜찮습니다. 그러나, 최대한 효율적이고 깔끔하게 구현해 보시길 바랍니다.
- 제출시 코드에 주석을 달아서 설명해야 하나요?  
아닙니다.
- 보고서에는 어떤 내용이 들어가야 하나요?  
전반적인 구현 방법과 구현에 대한 특이사항이 있으면 넣어주세요. 억지로 분량(2 장)을 채우실 필요는 없습니다.