실습 6 – 연결 리스트의 기본 연산

- 실습 목표
  - 연결 리스트의 기본 연산(insert, append, delete, search)을 이해하고 구현할 수 있다.
  - 연결 리스트를 이용하여 스택과 큐를 구현할 수 있다.

연결 리스트의 기본 연산(1)

- 리스트 정의
  - struct node {int data; struct node *next;} 구조 정의
  - struct node의 포인터로 *A, *temp, *ptr을 사용
    - A = NULL; 로 초기화

(6.1.)

- void insert_front(A, int data)
  - A 리스트의 제일 앞에 data를 갖는 새로운 노드 추가

- void print_list(A)
  - A 리스트의 내용을 앞에서부터 차례대로 출력

- void insert_last(A, int data)
  - A 리스트의 제일 뒤에 data를 갖는 새로운 노드 추가

연결 리스트의 기본 연산(2)

- int search(A, int data)
  - A 리스트에서 몇 번째 노드에 data가 저장되어 있는지를 출력 (노드 번호는 0부터 시작된다고 가정)
- int delete_front(A)
  - A 리스트에서 첫번째 노드를 삭제하고, 그 노드에 저장된 data를 return
- int delete(A, int data)
  - A 리스트에서 data가 저장된 노드를 삭제

연결 리스트를 이용한 스택과 큐의 구현

(6.2.)

- 스택
  - A 리스트에 데이터를 저장하고 삭제
  - insert_front를 이용하여 push() 구현
  - delete_front를 이용하여 pop() 구현

- 큐
  - A 리스트에 데이터를 저장하고 삭제
  - 제일 마지막 노드는 rear에서 저장
  - insert_last를 이용하여 add() 구현
  - delete_front를 이용하여 delete() 구현

추가 연산들

- 단순 연결 리스트를 역순으로 만드는 연산: invert()

- 두 개의 연결 리스트를 하나로 합치는 연산: merge()
  - 두 개의 연결 리스트 A와 B를 생성하는데, 각각 –1이 입력될 때까지 양수를 입력받아 오름차순으로 연결 리스트에 추가하라. 이후, **A, B를 하나의 리스트로 병합**하는데 역시 데이터 크기의 오름차순으로 병합한다.

- 원형 연결 리스트의 구현?