1 순차자료구조 개념과 선형자료구조 리스트의 이해

1 자료의 개념

- 자료는 구조화하는 방법에 따라
  리스트, 스택, 큐 데크, 트리 그래프 등으로 나뉨

- 이러한 자료구조 유형을 프로그램으로 구현하는
  방식에는 순차 자료구조와 연결자료구조가 있음

2 순차 자료구조의 개념

- 구현할 자료들을 논리적 순서로
  메모리에 연속 저장하는 구현 방식
- 논리적인 순서와 물리적인 순서가 항상 일치해야 함
- C 프로그래밍에서 순차 자료구조의
  구현 방식 제공하는 프로그램 기법은 배열

2 순차 자료구조의 개념

구분	순차 자료구조	연결 자료구조
메모리 저장 방식	메모리의 저장 시작 위치부터 빈자리 없이 자료를 순서대로 연속하여 저장한다. 논리적인 순서와 물리적인 순서가 일치하는 구현 방식이다.	메모리에 저장된 물리적 위치나 물리적 순서와 상관없이, 링크에 의해 논리적인 순서를 표현하는 구현 방식이다.
연산 특징	삽입·삭제 연산을 해도 빈자리 없이 자료가 순서대로 연속하여 저장된다. 변경된 논리적인 순서와 저장된 물리적인 순서가 일치한다.	삽입·삭제 연산을 하여 논리적인 순서가 변경되어도, 링크 정보만 변경되고 물리적 순서는 변경되지 않는다.
프로그램 기법	배열을 이용한 구현	포인터를 이용한 구현

[순차 자료구조와 연결 자료구조의 비교]

3 선형 리스트

선형리스트의 표현

- 자료의 특징(추상 자료형)과
  주로 사용할 연산(알고리즘)에 따라
  최적의 형태로 자료를 구조화해야 함
  - 자료를 구조화하는 가장 기본적인 방법 : **나열**

3 선형 리스트

선형리스트의 표현

- 리스트
  - 동창이름, 좋아하는 음식, 오늘 할일 등 하나씩 나열할 수 있는데 이렇게 나열한 목록을 말함

동창 이름	좋아하는 음식	오늘 할 일
상원	김치찌개	운동
상범	닭볶음탕	자료구조 스터디
수영	된장찌개	과제 제출
현정	잡채	동아리 공연 연습
…	…	…

[리스트 예]

3 선형 리스트

선형 리스트(Linear List)

- 원소들을 순서대로 나열한 리스트
  즉, 자료들 간에 순서를 갖는 리스트
  - 순서 리스트(Ordered List) 라고도 함

동창이름	
1	상원
2	상범
3	수영
4	현정
…	…

좋아하는 음식	
1	김치찌개
2	닭볶음탕
3	된장찌개
4	잡채
…	…

오늘 할 일	
1	운동
2	자료구조 스터디
3	과제 제출
4	동아리 공연 연습
…	…

[선형리스트 예]

4 리스트의 표현 형식

- 원소가 순서대로 나열된 선형리스트는
(a)에 나열된 순서가 원소들의 순서가 됨
- (b)는 동창이름 선형리스트를
리스트의 일반화 표현 형식으로 나타낸 것

[리스트 표현 형식과 예]

리스트 이름 = (원소 1, 원소 2, ⋯, 원소 n)
❶

(a) 리스트 표현 형식

동창 = (상원, 상범, 수영, 현정)

(b) 리스트 표현 예

4 리스트의 표현 형식

- 공백리스트
  - 원소가 하나도 없는 리스트

[공백 리스트 형식]

```
공백 리스트 이름 = ( )
```

2 선형 리스트의 저장, 삽입, 삭제

1 선형 리스트의 저장

선형리스트

- 순차 방식으로 구현하는 선형 순차 리스트

> 순차 자료구조는 원소를 논리적인 순서대로 메모리에 연속하여 저장

연결 리스트

- 연결 방식으로 구현하는 선형 연결 리스트

1 선형 리스트의 저장

선형 리스트의
메모리 저장 구조 예

선형 리스트에서
원소의 위치

2 선형 리스트에서 원소 삽입

- 선형리스트 중간에 원소가 삽입되면, 그 이후의 원소들은 한 자리씩 자리를 뒤로 이동하여 물리적 순서를 논리적 순서와 일치시킴

놀이공원에서 놀이기구를 타려고
줄을 서있다고 가정하면

▼

누군가 중간에 새치기를 하면
새치기 한 사람 뒤에 있는 사람은
모두 한자리씩 뒤로 밀리게 됨

2 선형 리스트에서 원소 삽입

- 선형리스트는 순차 자료구조 방식으로 구현
  - 삽입 후에 변경된 논리적 순서와 메모리에 연속 저장된 물리적 순서가 일치해야 함
  - 따라서 메모리에 순서대로 연속 저장되어 있는 선형 리스트에 새로운 원소를 삽입하려면 먼저 물리적으로 삽입할 자리를 만든 후에 원소를 삽입해야 함
  - 원소를 삽입할 자리를 만들려면 삽입할 위치 다음에 잇는 원소를 모두 한자리씩 뒤로 옮겨야 함

2 선형 리스트에서 원소 삽입

[새치기 전과 후의 위치와 순서 변화]

2 선형 리스트에서 원소 삽입

원소 삽입 방법

- 원소를 삽입할 **빈 자리 만들기**
  - 삽입할 자리 이후의 원소들을 **한 자리씩 뒤로 자리 이동**
- 준비한 빈 자리에 **원소 삽입하기**

2 선형 리스트에서 원소 삽입

삽입할 자리를 만들기 위한 자리 이동 횟수

- (n+1)개의 원소로 이루어진
  선형 리스트에서 k번 자리에 원소를 삽입하는 경우
  - k번 원소부터 마지막 인덱스 n번 원소까지
    (n-k+1)개의 원소를 이동

- 이동횟수
  = n-k+1
  = 마지막 원소의 인덱스 - 삽입할 자리의 인덱스 +1

2 선형 리스트에서 원소 삽입

삽입할 자리를 만들기 위한 자리 이동 횟수

2 선형 리스트에서 원소 삽입

삽입할 자리를 만들기 위한 자리 이동 횟수

- (n+1)개의 원소로 이루어진
  선형 리스트에서 k번 자리에 원소를 삽입하는 경우
  - k번 원소부터 마지막 인덱스 n번 원소까지
    (n-k+1)개의 원소를 이동

- 이동횟수
  = n-k+1
  = 마지막 원소의 인덱스 - 삽입할 자리의 인덱스 +1

3 선형 리스트에서 원소 삭제

놀이공원 줄서기에서
중간에 누군가 빠지게 되면

▼

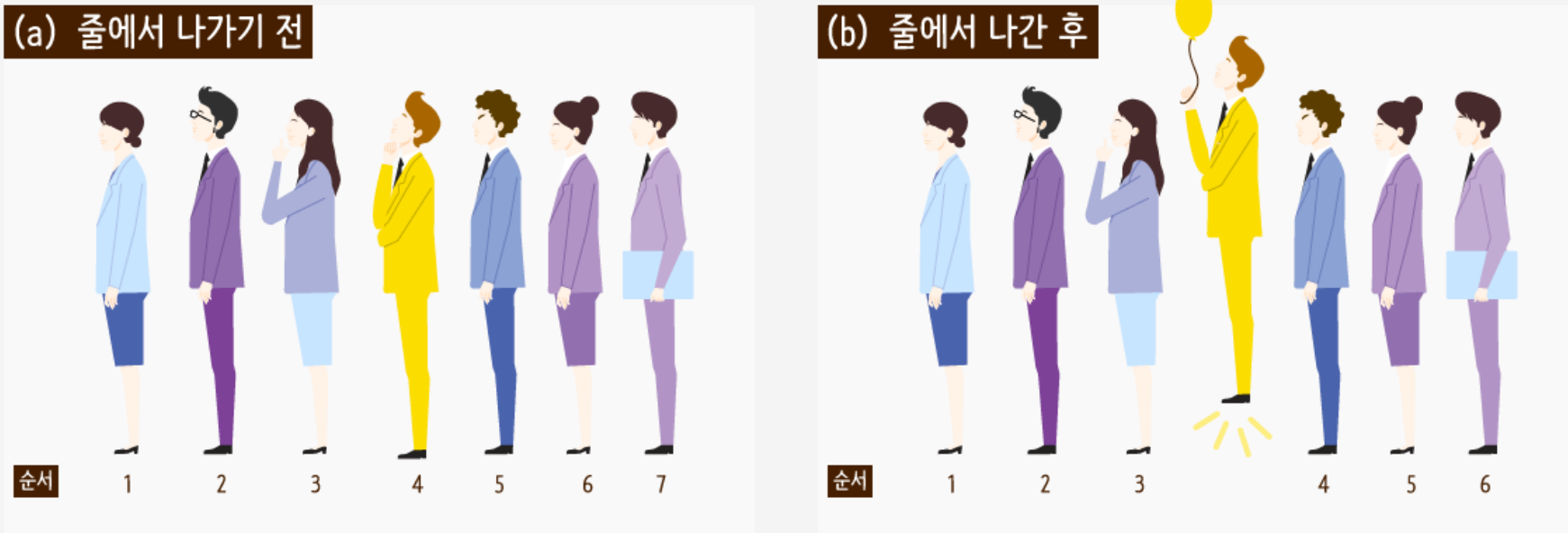
빠져나간 사람 뒤에 있던 사람은
모두 한자리씩 앞으로 당겨진다

- 선형리스트 중간에서 원소가 삭제되면, 원소가 있던 자리는 빈자리가 됨

3 선형 리스트에서 원소 삭제

- 선형리스트는 원소를 논리순서와 같은 순서로 메모리에 연속하여 저장해야 하는 순차구조이므로 빈자리가 없어야 함
  - 따라서, 그 이후의 원소들은 한 자리씩 자리를 앞으로 이동하여 물리적 순서를 논리적 순서와 일치시킴

3 선형 리스트에서 원소 삭제



[줄에서 사람이 나간 후의 위치와 순서 변화]

3 선형 리스트에서 원소 삭제

원소 삭제 방법

- 원소 삭제하기
- 삭제한 빈 자리 채우기
  - 삭제한 자리 이후의 원소들을 한자리씩 앞으로 자리 이동

3 선형 리스트에서 원소 삭제

삭제 후, 빈 자리를 채우기 위한 자리이동 횟수

- (n+1)개의 원소로 이루어진
  선형 리스트에서 k번 자리의 원소를 삭제한 경우
  - (k+1)번 원소부터
    마지막 n번 원소까지 (n-(k+1)+1)개의 원소를 이동

- 이동횟수
  = n-(k+1)+1
  = n-k
  = 마지막 원소의 인덱스-삭제한 자리의 인덱스

3 선형 리스트에서 원소 삭제

삭제 후, 빈 자리를 채우기 위한 자리이동 횟수

[선형 리스트에서의 원소 삭제]

3 1차원 배열을 이용한 선형리스트 구현

3 | 1차원 배열을 이용한 선형리스트 구현

1 선형 리스트의 구현

- 배열은 순서를 가진 배열 원소들을 메모리에 연속하여 순차적으로 구성
- 프로그래밍 언어에서 제공하는 배열을 사용하면 순차 자료구조 방식의 선형 리스트를 쉽게 구현할 수 있음

3 | 1차원 배열을 이용한 선형리스트 구현

1 선형 리스트의 구현

순차 구조의 배열을 사용

- **배열** : <인덱스, 원소>의 순서쌍의 집합
- **배열의 인덱스** : 배열 원소의 순서 표현

> - **1차원 배열**
> 인덱스를 하나만 사용하는 배열
> - 원소 순서를 한 개의 값으로
> 구별할 수 있는 간단한 선형 리스트는
> 1차원 배열을 사용하여 표현할 수 있음

2 1차원 배열을 이용한 선형 리스트 구현

- 1차원 배열을 이용한 구현
  - 분기별로 판매량을 순서대로 관리해야 하므로 선형리스트로 표현 할 수 있음

분기	1/4분기	2/4분기	3/4분기	4/4분기
판매량	157	209	251	312

[분기별 노트북 판매량 리스트]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 1차원 배열을 이용한 선형 리스트 구현

[분기별 판매량 선형 리스트 예]

```
int sale[4]={157, 209, 251, 312};
```

(A) 분기별 판매량 선형 리스트의 1차원 배열 선언

	[0]	[1]	[2]	[3]
sale	157	209	251	312

(a) 분기별 판매량 선형 리스트의 논리적 구조

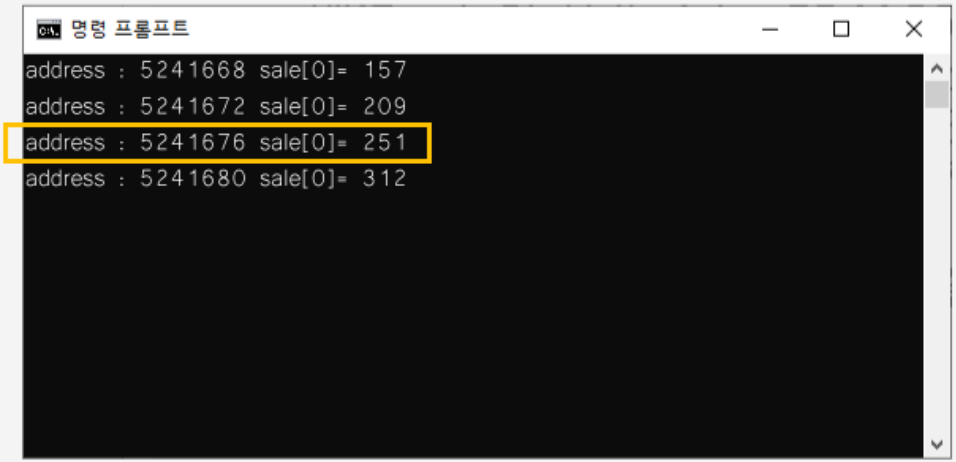
(b) 분기별 판매량 선형 리스트의 물리적 구조

3 원소의 논리적·물리적 순서 확인하기 프로그램

예제 원소의 논리적·물리적 순서 확인하기

```
01  #include <stdio.h>
02
03  void main() {
04                  int i, sale[4]={157, 209, 251, 312};
05
06                  for (i=0; 1<4; i++){
07                                  printf("\n adress:%u sale[%d]=%d", &sale[i], i, sale[i]);
08                  }
09
10                  getchar();
11  }
```

4 실행 결과



- 실행 결과 확인
  배열 sale의 시작 주소 : 5241668
  sale[2]=251의 위치 = 시작 주소 + (인덱스 x 4바이트)
  = 5241668 + (2 x 4바이트)
  = 5241676
  - **논리적인 순서대로 메모리에 연속하여 저장된 순차 구조임을 확인!**