

# 과제 1. 배열

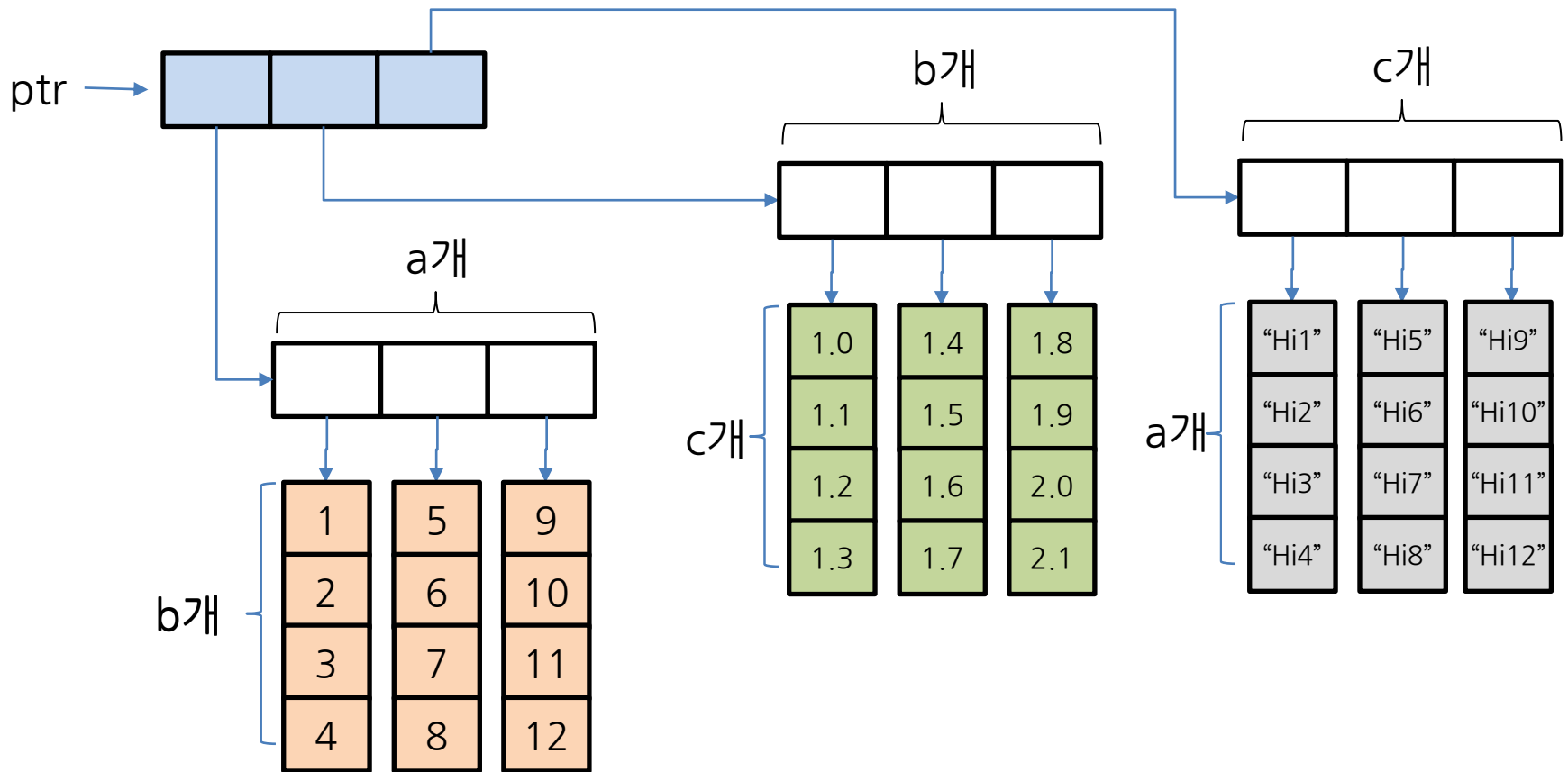
# 과제 1

- ◆ 제출 마감일 : 4월 7일 (일요일) **10:59pm**
- ◆ eClass 과제방에 제출
- ◆ 개인 과제
- ◆ 사용 언어
  - C (C++ 도 가능)
- ◆ 제출 양식
  - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
    - 보고서에는 소스코드를 넣지 말 것. 그림과 설명으로 작성할 것.
  - 소스코드 (**별도 파일로 제출**)
- ◆ 보고서는 PDF 로 제출

타인의 과제를 복사하지 말 것

# (1) 자료구조 만들기

- ◆ 사용자에게 숫자  $a$ ,  $b$ ,  $c$ 를 입력 받은 후, 다음 모양을 가진 자료구조를 만들어라.



# (1) 자료구조 만들기

## ◆ 고려할 점

- 사용자가 입력한  $a, b, c$ 의 숫자에 맞추어 공간을 할당해야 한다.
- 사각형 안에 써진 숫자나 문자는 그 안에 넣어야 하는 값을 의미한다.
- 모든 공간은 ptr에서 시작된다.
- 모든 공간은 동적 할당해서 사용해야 한다.  $a, b, c$ 가 가변적이므로 동적 할당할 수 밖에 없다.
- 앞 그림의 파란색 공간은 3개로 결정되어 있지만, 이 부분도 동적 할당한다.
- 첫 번째 공간(붉은 색)은 정수가, 두 번째 공간(녹색)은 실수가, 세 번째 공간(회색)은 문자열이 저장된다.

# (1) 자료구조 만들기

## ◆ 프로그램 실행 결과

- 사용자에게 a, b, c를 입력 받은 후,
- 생성한 공간에 정해진 값을 채우고
- 모든 공간에 있는 값을 화면에 표시한다.
  - 값을 화면에 표시할 때에는 해당 공간에 찾아가 값을 표시해야 한다. 결과만 동일하게 그려서는 안된다.
  - 값을 화면에 표시할 때에는 반드시 포인터 변수 ptr을 이용하여 원하는 공간에 찾아가야 한다.
- 결과만 동일하게 표시해서는 안되고, 자료구조가 동일하게 생성되어야 한다.

# (1) 자료구조 만들기

## ◆ 실행 예)

a, b, c? 2 3 2

1 2 3

4 5 6

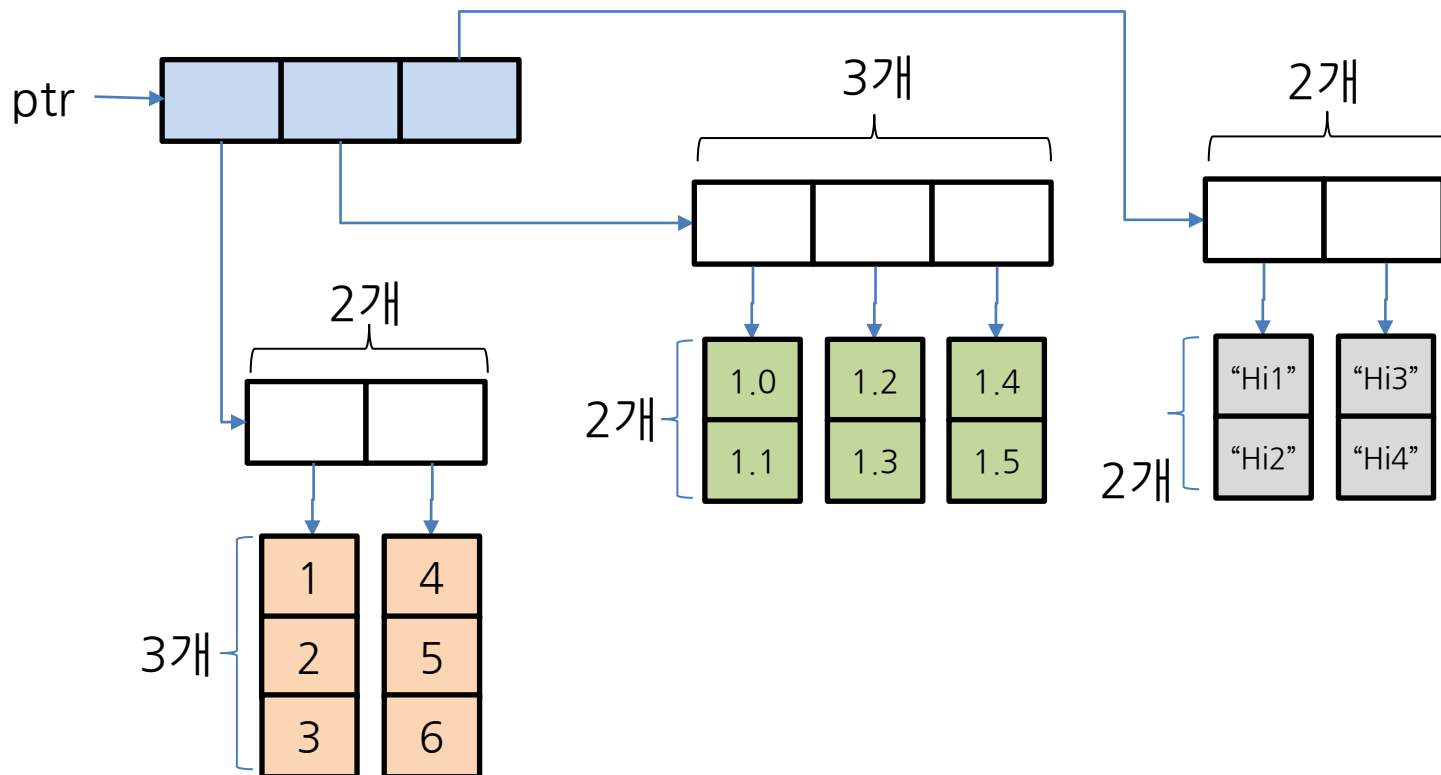
1.0 1.1

1.2 1.3

1.4 1.5

hi1 hi2

hi3 hi4



## (2) 파이썬의 List 내부 구조 파악하기

### ◆ 파이썬의 List의 내부 구조를 확인한다.

- list는 insert, delete 가 용이하며, 다양한 자료형의 데이터를 담을 수도 있다.
- 파이썬은 C/C++로 작성되었으므로, 우리가 파이썬의 List의 구조를 알 수 있으면 List와 동일한 구조를 C 언어에서도 사용할 수 있을 것이다.

### ◆ 파이썬의 List는 어떻게 구현되어 있는지 문서로 정리한다.

- 이 구조의 장점이 무엇인지 설명한다.

List의 내부 구조를 정리한 자료들이 있으니 그것을 참고할 것(파이썬 소스코드를 직접 보는 것은 내용이 방대하여 권장하지 않음)

# (3) 다항식의 저장 방식 개선하기

## ◆ 다항식 저장 방식 개선하기

- 강의자료에서는 다항식 저장 방식(2)에서도 미리 충분한 개수의 항의 수를 MAX\_DEGREE로 정의하여 사용하였다.
  - 몇 개의 항이 필요한지 모르는 상태에서 최대치를 정의하고 그 만큼의 공간을 할당하는 것은 비효율적이다.
- 좀 더 효율적인 다항식의 저장 방식이 되려면 어떤 식으로 개선하면 좋을지 설명하고,
- 이를 구현하여 다항식의 덧셈과 곱셈이 진행될 수 있게 하라.
  - 곱셈은 강의자료에 나와 있지 않음



### (3) 다항식의 저장 방식 개선하기

#### ◆ 실행 예)

– 1번 다항식 : 8 3 7 1 1 0

– 2번 다항식 : 10 3 3 2 1 0

–  $1 + 2 = 18x^3 + 3x^2 + 7x + 1$

–  $1 * 2 = 80x^6 + 24x^5 + 70x^4 + 39x^3 + 3x^2 + 7x + 1$

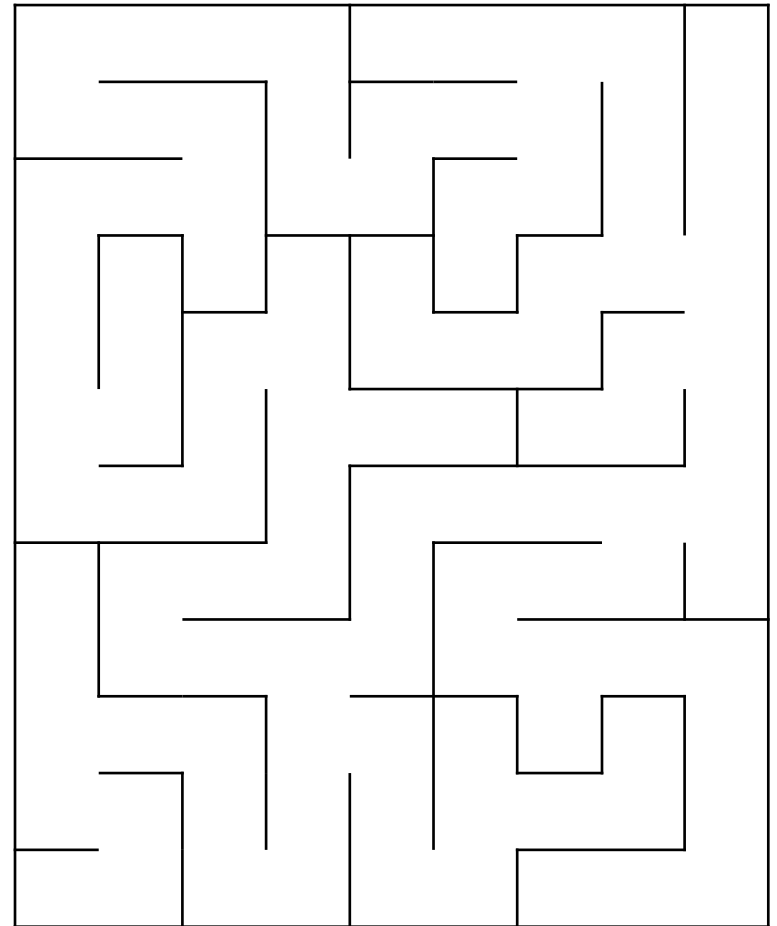
## (4) 희소 행렬 저장 방식의 곱셈

- ◆ 강의 자료에는 희소행렬 저장 방식의 덧셈이 나와 있다.
- ◆ 희소 행렬 저장 방식의 곱셈을 구현하라.
  - 희소 행렬 저장 방식으로 구현된 두 개의 행렬을 곱한 결과를 만드는 함수 `sparse_matrix_mult`를 구현하고, 이를 이용하여 행렬의 곱셈 결과를 화면에 표시하라.
  - 입력, 출력 양식은 각자가 정의하라.

## (5) 미로 표현하기

◆ 다음과 같이 미로가 파일로 작성되어 있다.

- **공간의 크기를 최소화**하여 미로를 메모리에 저장하는 방법을 제안하라.
- 예를 들어 maze1.txt 파일은 531 바이트이며, 미로 데이터를 메모리의 최소 크기 공간에 저장하는 새로운 방법은 531 바이트보다 작아질 것이다.
- 미로 데이터는 다음 파일을 확인
  - maze1.txt



## (5) 미로 표현하기

### ◆ 좋은 결과

- maze1 저장에 200 바이트 이내의 메모리 사용

### ◆ 아주 좋은 결과

- 미로1 저장에 50 바이트 이내의 메모리 사용

### ◆ 나쁜 결과

- 미로1 저장에 500 바이트 이상의 메모리 사용
- 저장 후에 원래의 미로로 복원이 안되는 경우

### ◆ 이 문제는 자신의 방법을 설명하는 것이며, 구현할 필요는 없다.