

# 과제 1

-메모리 관리-

전자공학 프로그래밍

# 과제1. 메모리 관리

---

- 과제 설명

# 과제 1. 메모리 관리

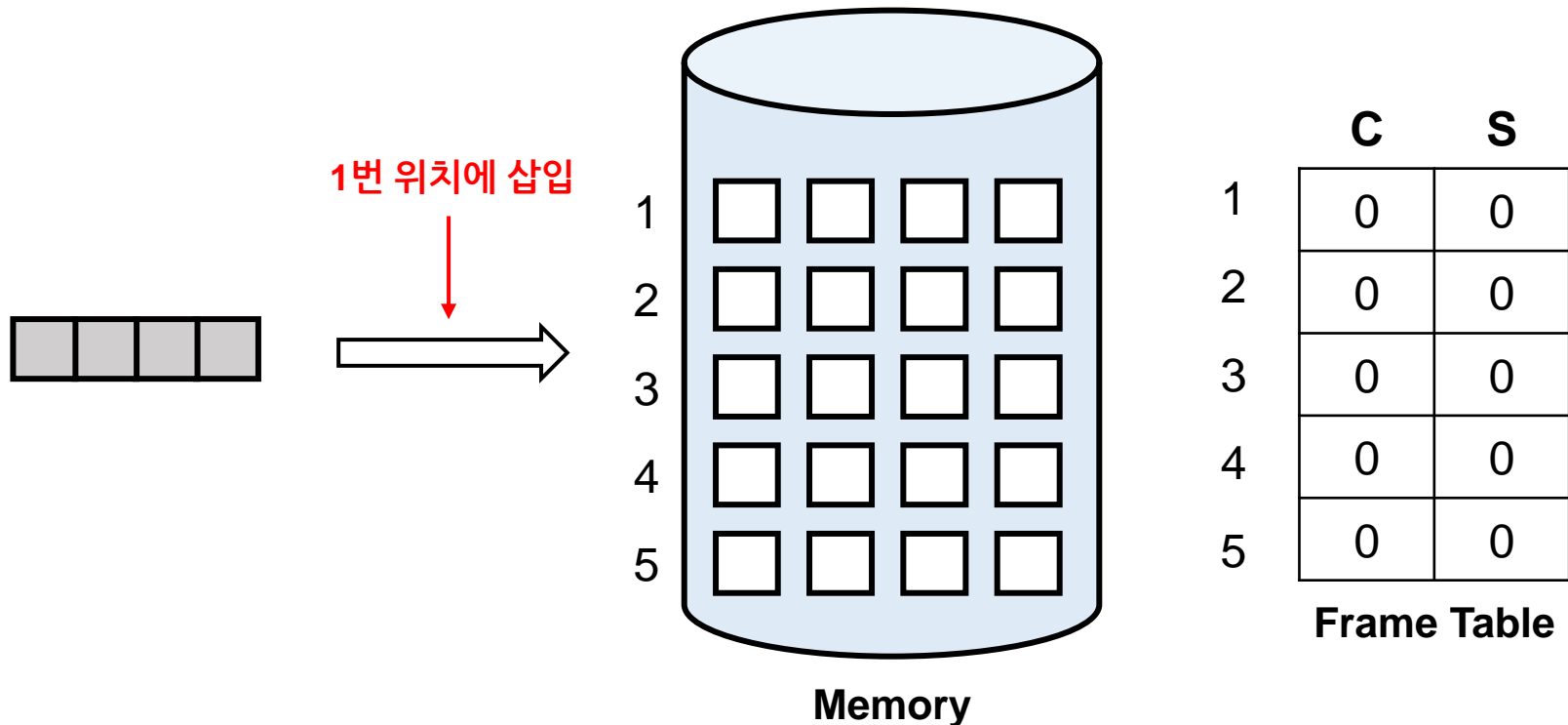
- 메모리 관리(Memory Allocation)
  - 목표 : 한정된 메모리 자원을 관리한다.
  - 메모리 관리 방법
    - ▶ 명령어를 입력받아 아래 3가지 메모리 관리 기능 수행
    - ▶ 1) 메모리 할당 (실습 내용)
      - 데이터를 메모리(배열)의 원하는 위치에 할당한다.
        - 입력된 데이터의 크기는 메모리 크기보다 작아야 함
        - 메모리 할당은 행(ROW) 단위로만 가능
        - 저장 가능한 공간이 있어야 할당
    - ▶ 2) 메모리 제거
      - 원하는 위치의 메모리에 저장된 데이터를 제거한다.
    - ▶ 3) 메모리 압축
      - 메모리를 압축하여 비어있는 메모리 공간을 제거한다.

# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (1) 메모리 할당 (명령어 : allocation)

- ▶ 실습 1 참고
- ▶ 데이터를 메모리(배열)의 원하는 위치에 할당한다.
- ▶ 데이터 크기 4, 메모리 1번 위치 할당

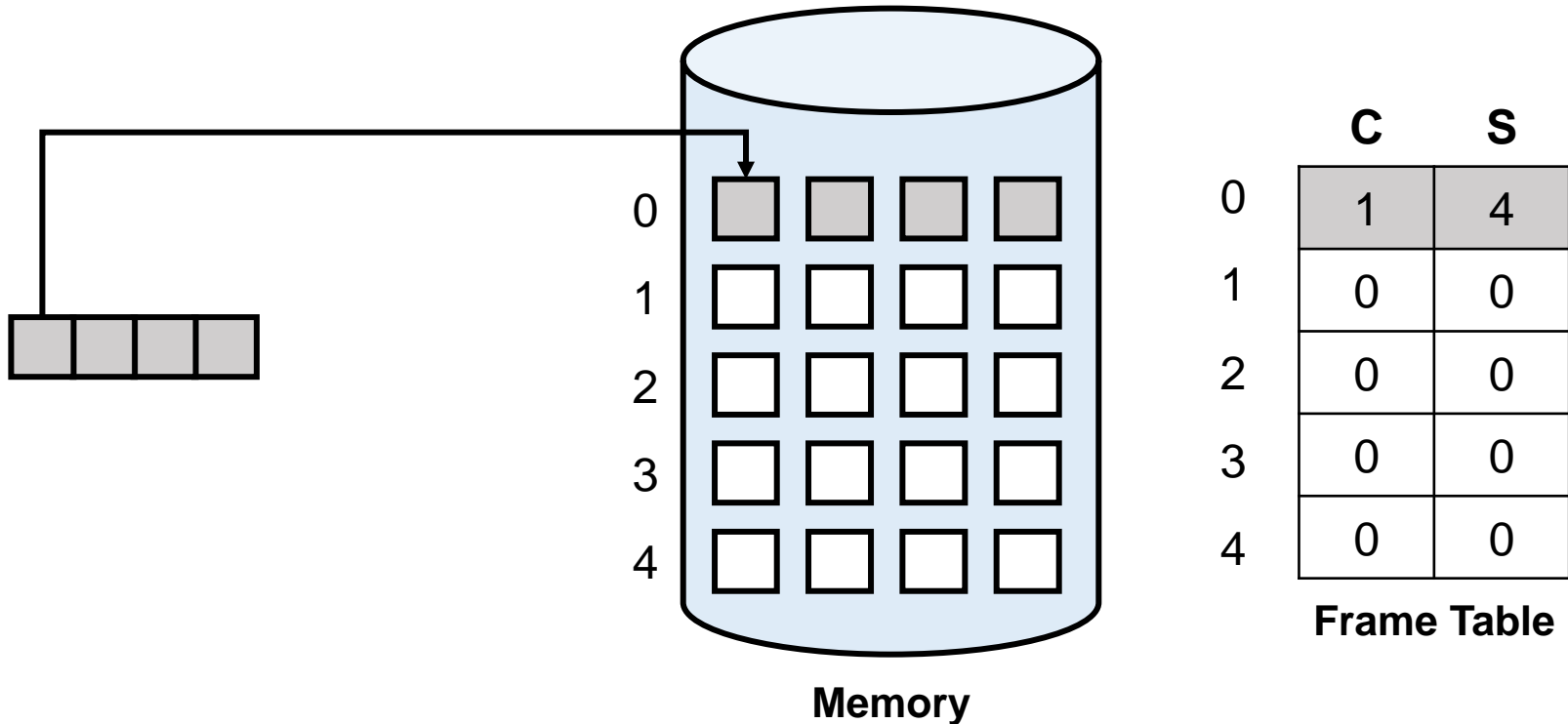


# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (1) 메모리 할당 (명령어 : allocation)

- ▶ 실습 1 참고
- ▶ 데이터를 메모리(배열)의 원하는 위치에 할당한다.
- ▶ 데이터 크기 4, 메모리 1번 위치 할당

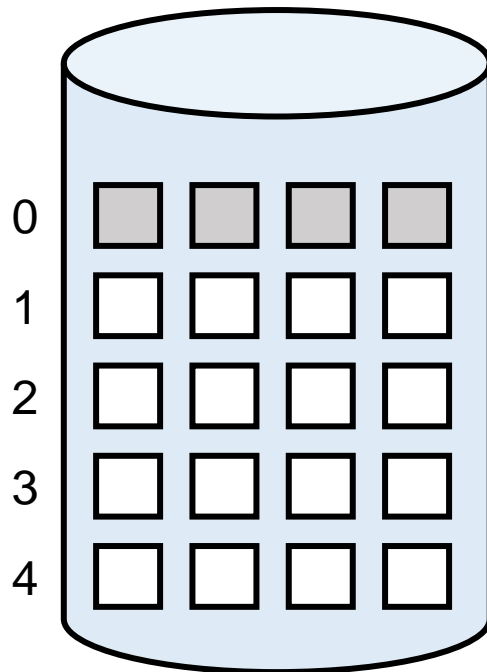


# 과제 1. 메모리 관리

- 메모리 관리

- (2) 메모리 제거 (명령어 : remove)

- 원하는 위치의 메모리에 저장된 데이터를 제거한다.
    - 0번 위치 메모리 제거



Memory

	C	S
0	1	4
1	0	0
2	0	0
3	0	0
4	0	0

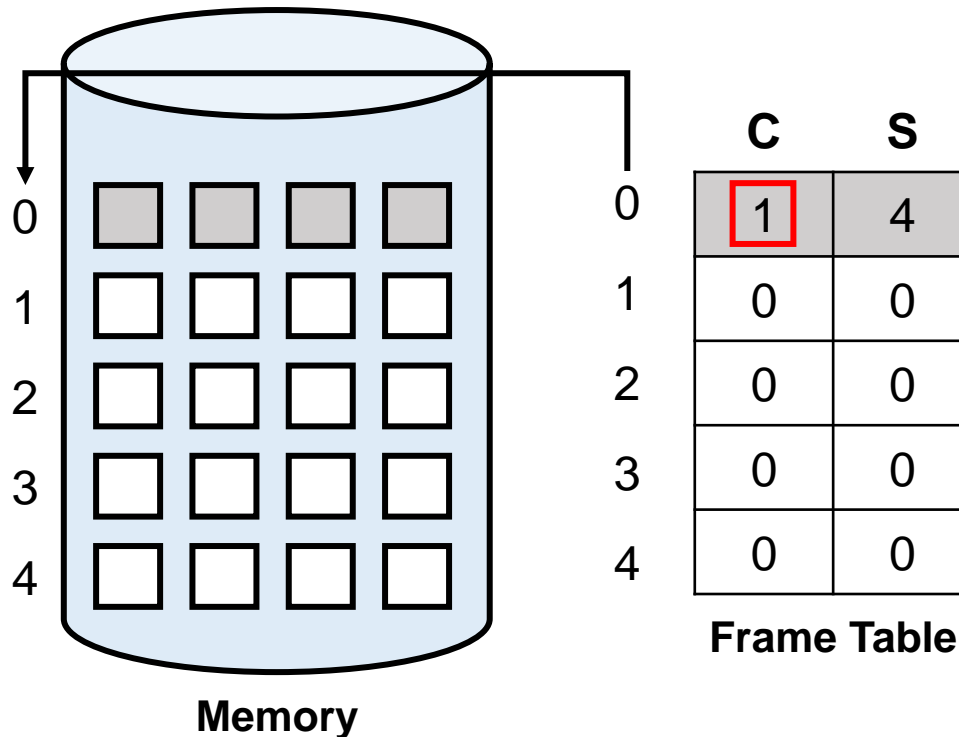
Frame Table

# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (2) 메모리 제거 (명령어 : remove)

- ▶ 원하는 위치의 메모리에 저장된 데이터를 제거한다.
- ▶ 0번 위치 메모리 제거
  - 프레임 테이블의 체크비트로 위치 확인후, 데이터 크기만큼 메모리 제거

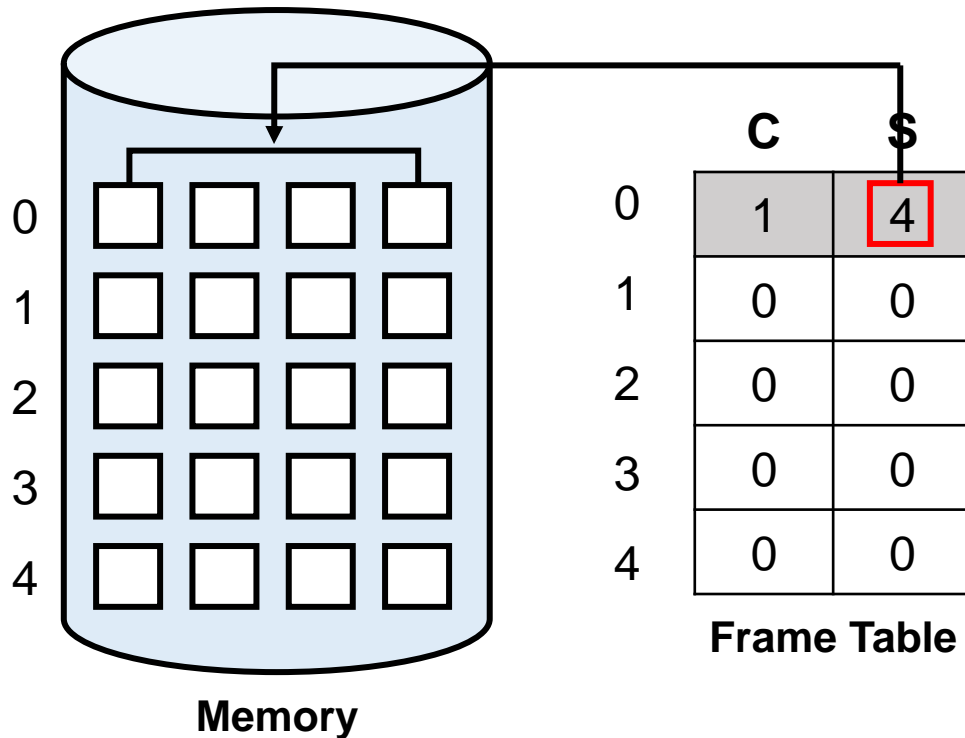


# 과제 1. 메모리 관리

- 메모리 관리

- (2) 메모리 제거 (명령어 : remove)

- 원하는 위치의 메모리에 저장된 데이터를 제거한다.
    - 0번 위치 메모리 제거
      - 프레임 테이블의 체크비트로 위치 확인후, 데이터 크기만큼 메모리 제거



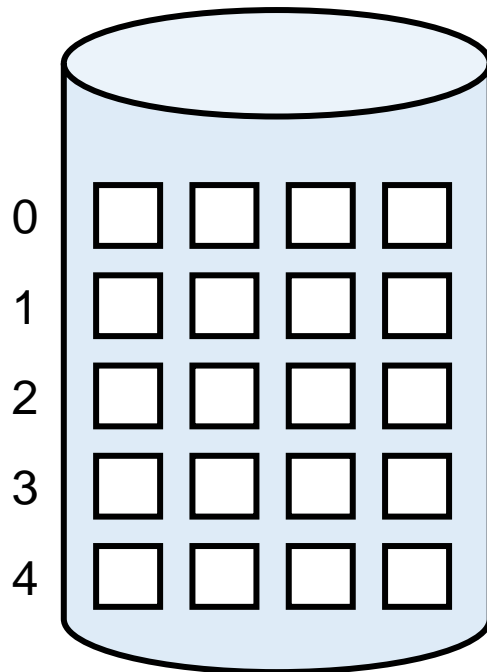


# 과제 1. 메모리 관리

- 메모리 관리

- (2) 메모리 제거 (명령어 : remove)

- ▶ 원하는 위치의 메모리에 저장된 데이터를 제거한다.
    - ▶ 0번 위치 메모리 제거
      - 프레임 테이블의 체크비트로 위치 확인후, 데이터 크기만큼 메모리 제거
      - 프레임 테이블 수정



Memory

	C	S
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

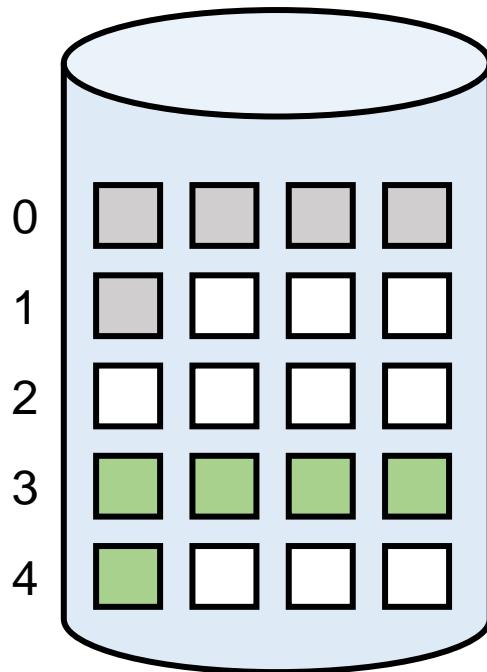
Frame Table

# 과제 1. 메모리 관리

- 메모리 관리

- (3) 메모리 압축 (명령어 : compaction)

- ▶ 메모리를 압축하여 비어있는 메모리 공간을 제거한다.



Memory

	C	S
0	1	5
1	0	0
2	0	0
3	1	5
4	0	0

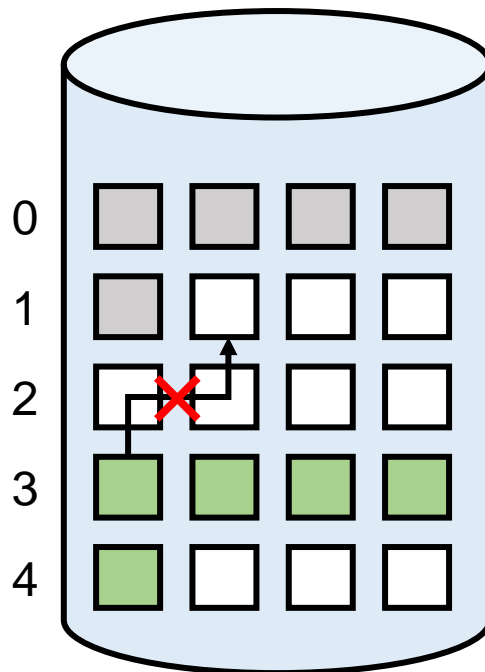
Frame Table

# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (3) 메모리 압축 (명령어 : compaction)

- ▶ 메모리를 압축하여 비어있는 메모리 공간을 제거한다.
- ▶ 메모리 공간을 확인하여 압축 가능할 경우 수행
  - 메모리는 행 단위로 할당되므로 1번의 나머지 부분에 할당 불가



Memory

	C	S
0	1	5
1	0	0
2	0	0
3	1	5
4	0	0

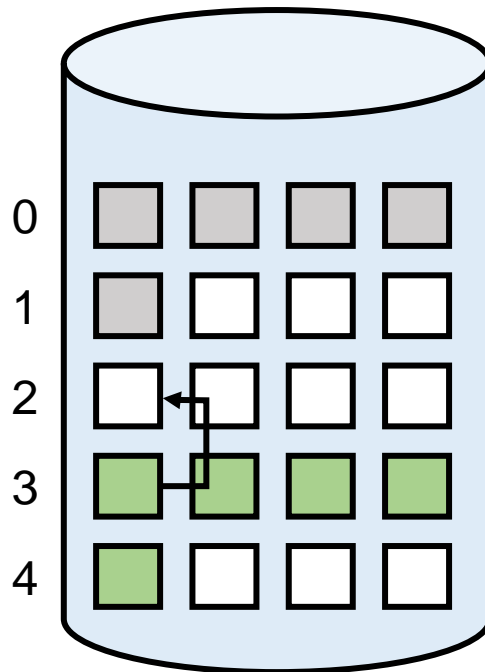
Frame Table

# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (3) 메모리 압축 (명령어 : compaction)

- ▶ 메모리를 압축하여 비어있는 메모리 공간을 제거한다.
- ▶ 메모리 공간을 확인하여 압축 가능할 경우 수행
  - 메모리는 행 단위로 할당되므로 1번의 나머지 부분에 할당 불가
  - 따라서 비어있는 2번 위치로 압축 및 프레임 테이블 수정



Memory

	C	S
0	1	5
1	0	0
2	0	0
3	1	5
4	0	0

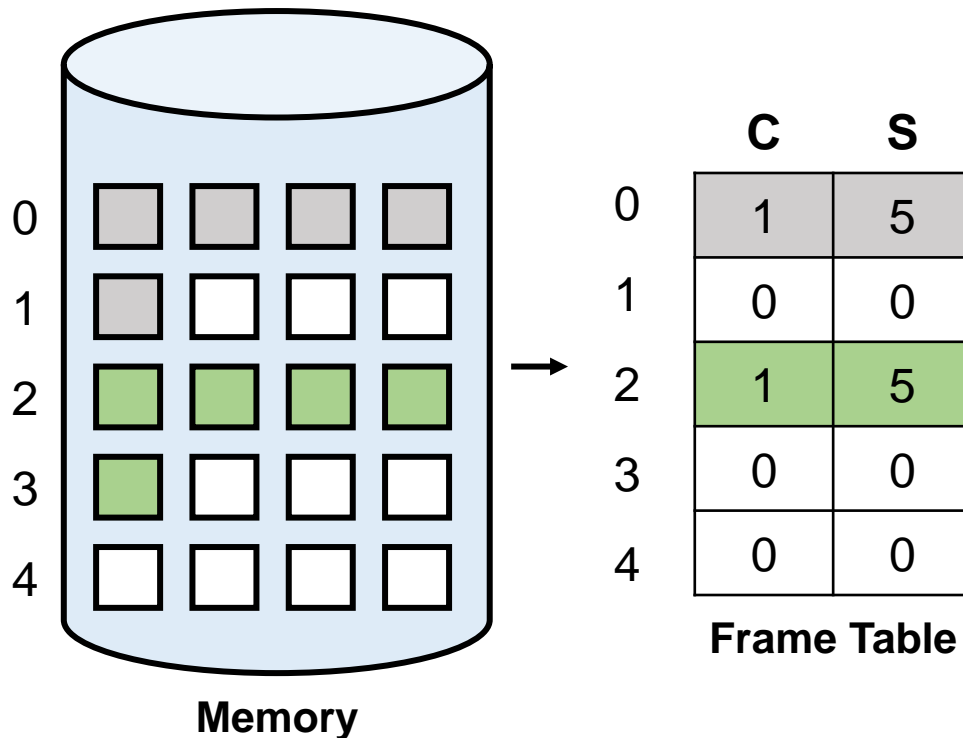
Frame Table

# 과제 1. 메모리 관리

## • 메모리 관리

### ■ (3) 메모리 압축 (명령어 : compaction)

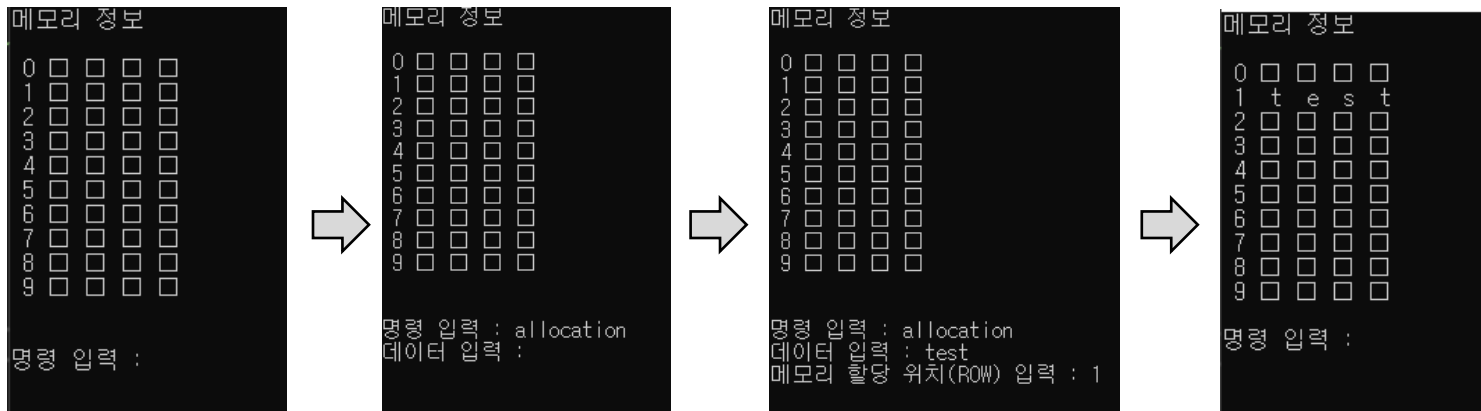
- ▶ 메모리를 압축하여 비어있는 메모리 공간을 제거한다.
- ▶ 메모리 공간을 확인하여 압축 가능할 경우 수행
  - 메모리는 행 단위로 할당되므로 1번의 나머지 부분에 할당 불가
  - 따라서 비어있는 2번 위치로 압축 및 프레임 테이블 수정



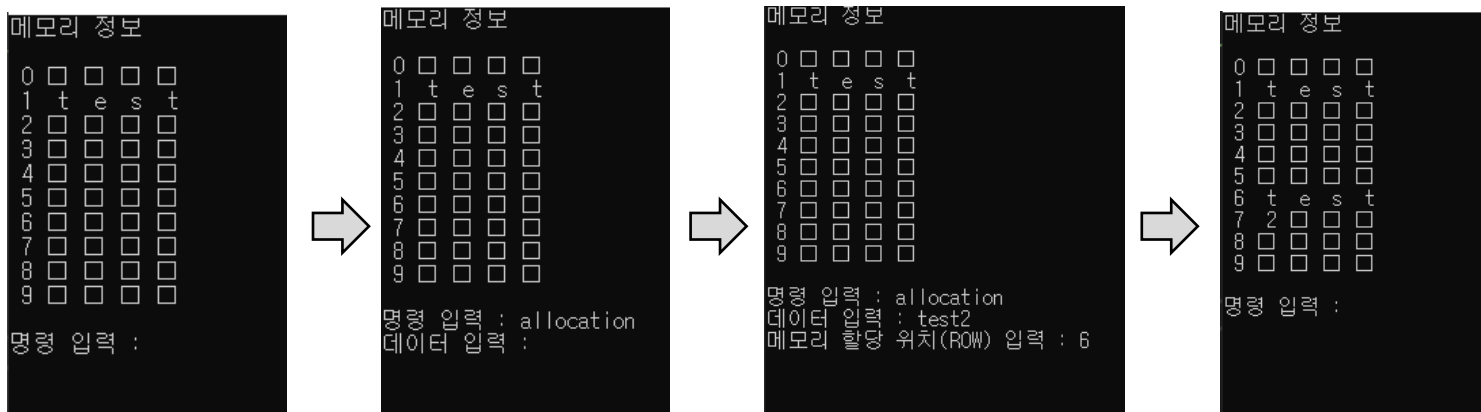
# 과제 1. 메모리 관리

## 구현 결과 예시(1)

### “test” 데이터에 대한 메모리 할당



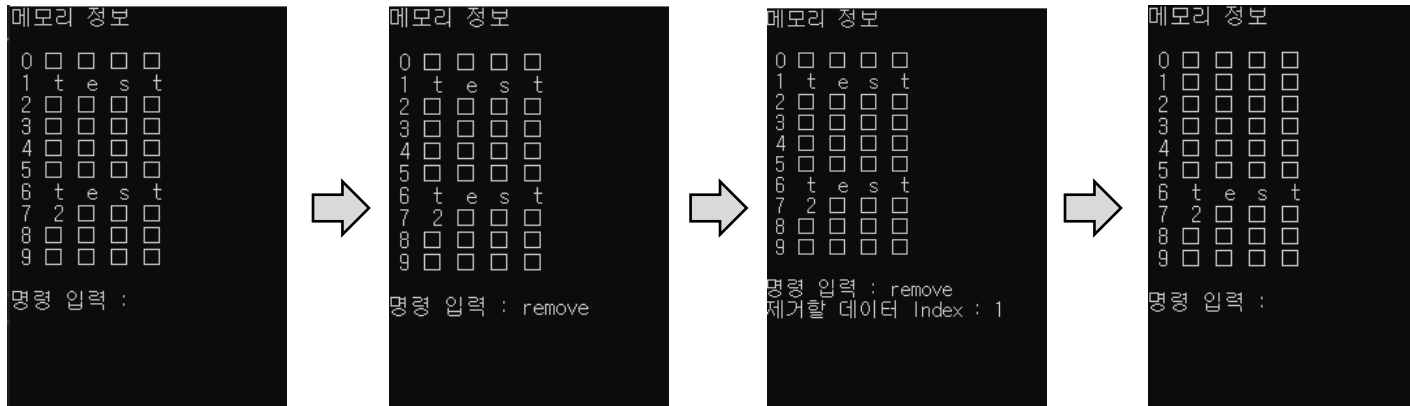
### “test2” 데이터에 대한 메모리 할당



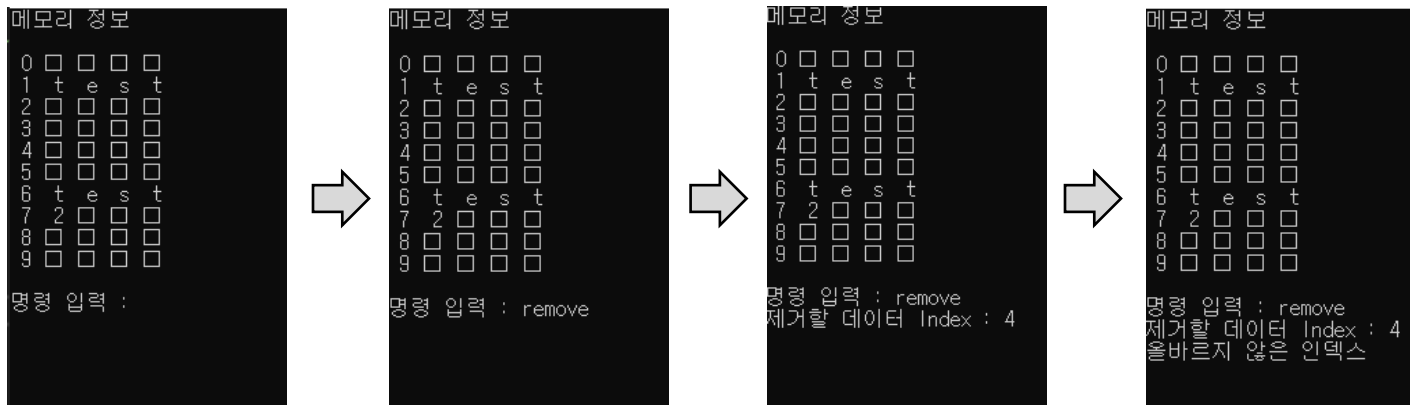
# 과제 1. 메모리 관리

## 구현 결과 예시(2)

### 1번 메모리 데이터 제거

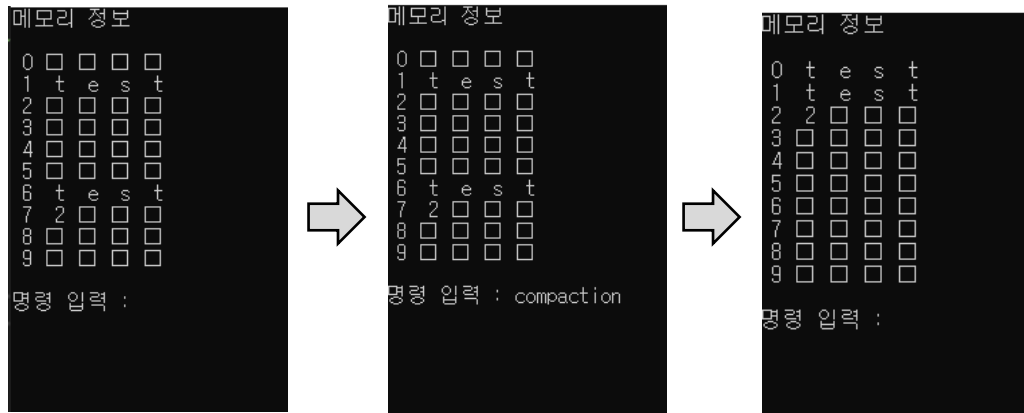


### 비어있는 메모리 데이터 제거

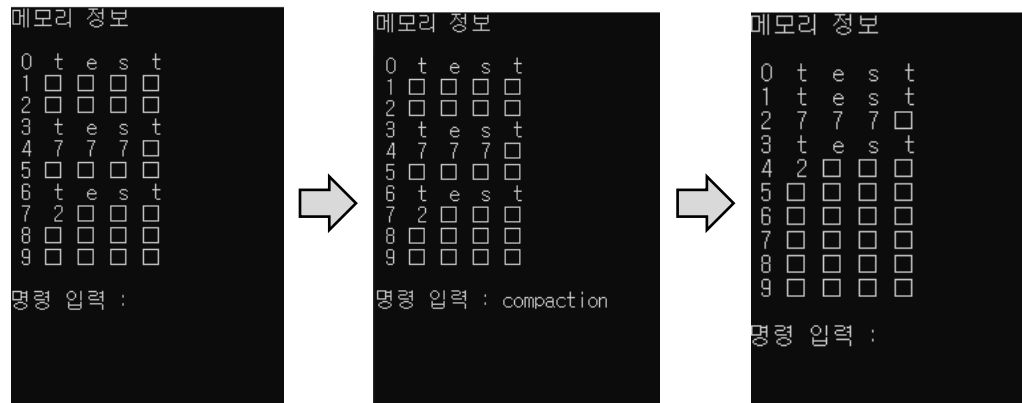


# 과제 1. 메모리 관리

- 구현 결과 예시(3)
  - 메모리 압축(1)



- 메모리 압축(2)





# 과제 채점 기준

- Due Date : 4/5(수) 자정 (기한 엄수)

감점	지각	(4/5에서 4/6로 넘어가는 자정까지 제출)	감점 x	블랙보드에 업로드 해주세요. (블랙보드가 문제되는 경우, 메일로 보내주세요.)
		~ 1일 지각	(총점수) * 0.75	총 점수에서 25% 감점
		~ 3일 지각	(총점수) * 0.50	총 점수에서 50% 감점
		3일 이상 지각	0	과제 마감 시점을 기준으로 3일 이후로는 받지 않습니다.
	코드 미제출	보고서 코드가 있을 때,	(보고서 점수) * 0.50	
		보고서에 코드가 없을 때,	구현점수 x	
	메인 함수 수정		상관 없음	
	cpp 파일 제출			c 파일로 확장자 변환 후 구현 채점기준 적용
	Copy		0	

# 과제 채점 기준

- 평가 기준

- <구현(50%) + 보고서(50%)>

: 구현과 보고서의 평가는 기능별로 평가를 할 예정이며, 구현의 경우는 각 기능을 명시된 설명에 맞게 구현하였는지, 구현하지 못하였다면 자신이 구현한 부분에 대해 어느 정도 설명이 되어 있는지를 보고 평가할 예정입니다

- ✓ **필수** : Visual Studio 2019로 구현 권장
  - ✓ 해당 환경에서 작동하지 않거나 오류 발생시 **불이익** 있을 수 있음
- ✓ 각 함수에 대한 설명은 주석을 참조하여 구현
- ✓ 결과 화면에 수행해 본 예시에 대한 분석 필수(보고서)
- ✓ 프로젝트 제출 x, .cpp 파일 x
  - ✓ 프로젝트 제출 또는 .cpp 파일 제출의 경우 **감점** 예정
- ✓ **Copy시 0점 처리는 물론, 추가 불이익이 있음**
  - ✓ 두 명의 조교가 line by line으로 채점할 예정

# 과제 채점 기준

- 평가 기준
  - 구현 부분(5점)
    - ▶ 구현 부분에 대한 **기능 평가** 예정
    - ▶ 메인 문 구현 1점
    - ▶ 메모리 제거 구현 2점
    - ▶ 메모리 압축 구현 2점
  - 보고서 부분(5점)
    - ▶ 구현된 함수의 기능 명세
    - ▶ 주어진 작성된 코드 분석, 작성한 소스코드 분석, 사용한 방식 설명
    - ▶ 결과 화면에 대한 분석
      - 모든 보고서 작성은 본인이 직접 작성한 내용이어야 함 (강의노트 paste X)
  - **합산 점수는 추후 과제 점수 비율에 따라 조정될 예정**
  - .c 파일과 보고서 파일 제출 (프로젝트 제출 x, .cpp 파일 x)

# Q&A

---

**AJOU UNIVERSITY**  
Embedded & Software Lab.