

DATA STRUCTURE

Week 2

- Dynamic memory allocation -



INDEX

I. Pointer Arithmetic (포인터 연산)

II. Dynamic Memory Allocation (동적 메모리 할당)

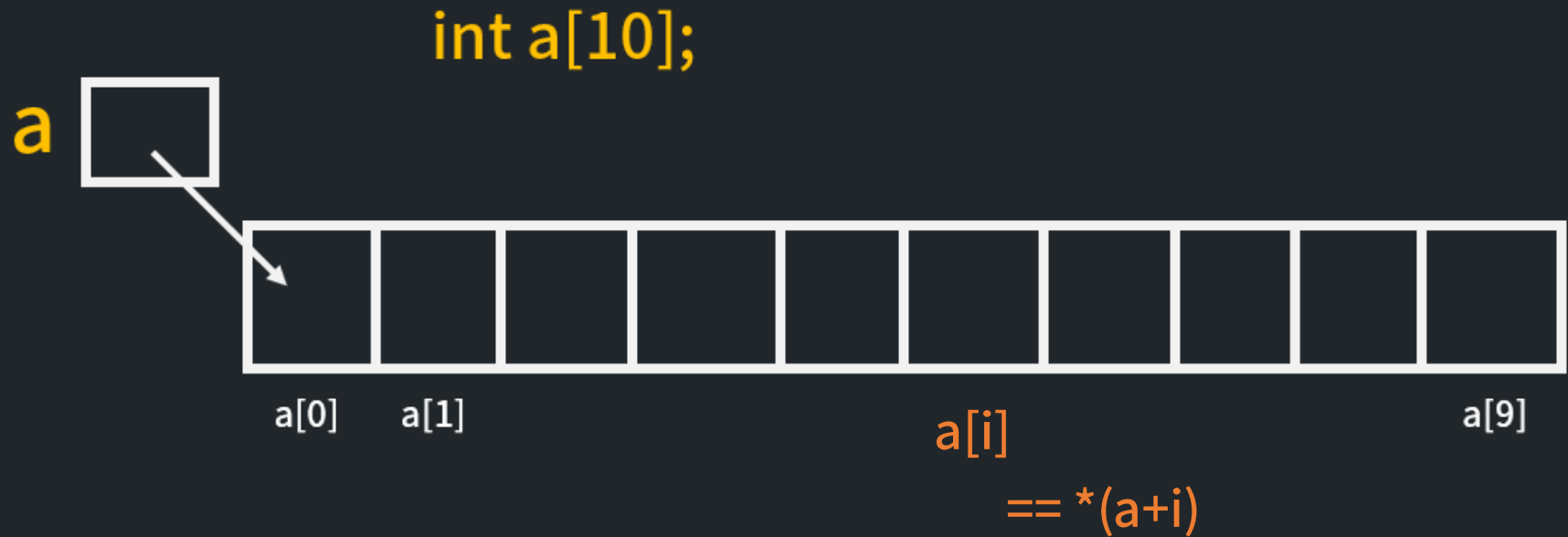
III. malloc function

IV. Array reallocation (배열 키우기)

V. Wrap up (실습)

I. Pointer Arithmetic

- $*a$ 와 $a[0]$ 은 동일한 의미이다.
- 또한 $a[1]$ 은 $*(a+1)$ 과 동일하고, $a[i]$ 는 $*(a+i)$ 와 동일하다.



이런 연산을 pointer arithmetic 이라고 부른다

II. Dynamic memory allocation

- 변수를 선언하는 대신 프로그램의 요청으로 메모리를 할당할 수 있다.
- 이것을 동적 메모리 할당 (dynamic memory allocation) 이라고 부른다.
- malloc 함수를 호출하여 동적 메모리 할당을 요청하면,
요구하는 크기의 메모리를 할당하고 그 시작 주소를 반환한다.

III. malloc 함수

```
int *p;
p = (int *) malloc (40);
if (p == NULL) {
    /* 동적 메모리 할당이 실패 */
    /* 적절한 조치를 취한다 */
}
```

malloc이 반환하는 주소는 타입이 없는 void이나,
정수들을 저장하기 위해서 이것을 int * 로 변환한다.

할당받을 메모리의 크기를 byte 단위로 지정한다.
여기서는 10개의 정수를 저장하기 위해서 40바이트를 요청

```
p[0] = 12;
p[1] = 24;
*(p+2) = 36;
```

malloc으로 할당받은 메모리는 이렇게 보통의 배열처럼 사용

IV. Array reallocation

- 동적으로 할당된 배열은 공간이 부족할 경우 더 큰 배열을 할당하여 사용할 수 있다.

```
int * array = (int *) malloc (4 * sizeof(int));
```

/ 배열 array의 크기가 부족한 경우가 발생한다 */*

```
int * tmp = (int *) malloc (8 * sizeof(int));
```

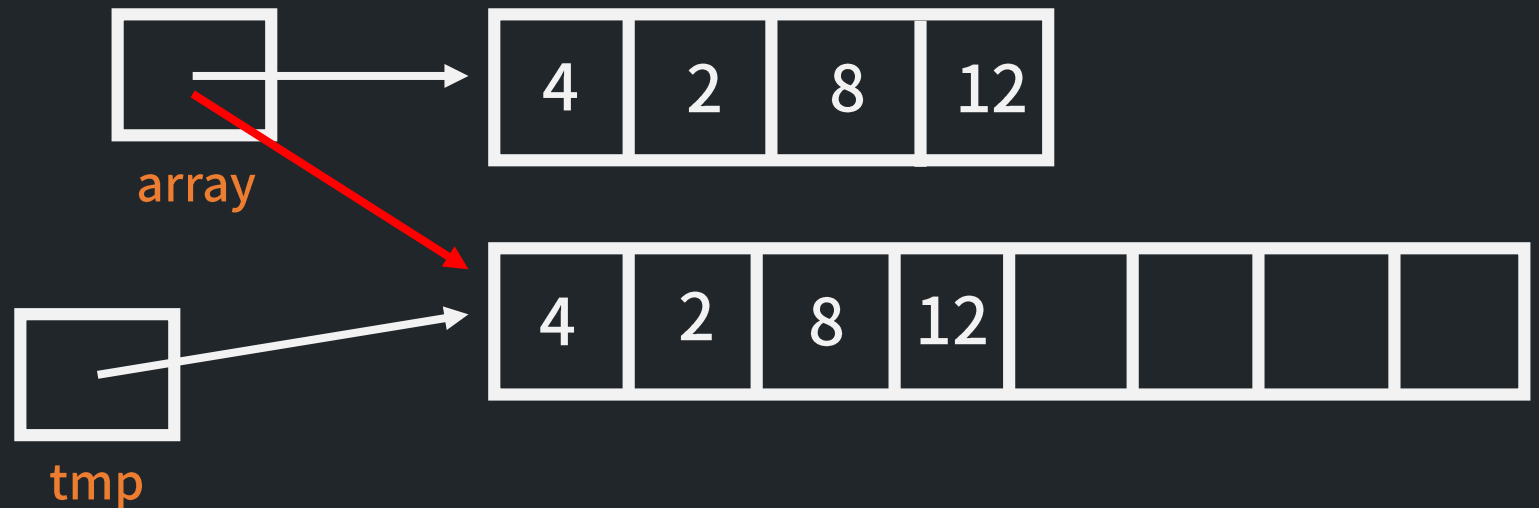
```
int i;
```

```
for (i=0; i<4; i++) {  
    tmp[i] = array[i];
```

```
}
```

```
array = tmp;
```

‘그럼 원래 할당 받았던 16바이트 공간은 어떻게 되는가?’



실습 문제 #1

N개의 정수를 입력 받아 동적 메모리 할당을 사용하여 배열에 저장한 후, 저장된 배열 요소들의 합을 출력하는 프로그램을 작성하시오.

입력예시

6 → N

3 2 0 1 4 6

출력예시

16

실습 문제 #2

다음과 같은 기능을 하는 학생번호 관리 프로그램을 작성하시오.

- 1) 등록할 학생번호 수 N 을 입력받고, N 개의 학생번호를 동적할당을 사용해 입력 (int형 사용)
- 2) 삭제할 학생번호 개수 D ($D < N$) 를 입력받고, D 만큼 메모리 크기 조정 후 출력
- 3) 마지막에 입력한 학생번호부터 D 개를 삭제한다

입력예시

3 → N

10113 → 1학년 1반 13번

21234

30307

2 → D

출력예시

10113

실습 문제 #3 배열과 포인터, 동적할당 총정리

Baekjoon Online Judge #4344 '평균은 넘겠지' 문제

정답률 37.56%

<https://www.acmicpc.net/problem/4344>

Thank You