

DATA STRUCTURE

Week 1

- Reviewing C -

김지오



INDEX

I. 수업을 듣기 전에


II. 사전 지식 점검 BY 간단한 문제풀이

III. Data Structure 란 무엇이고, 왜 필요한가?

IV. Reviewing C

I. 수업을 듣기 전에

 설명은 한국어로 / 주요 용어는 영어로 설명할 수 있도록 할 예정

 수업은 개념 설명 + 문제 풀이 (수업당 1~2문제 가량) 로 진행될 예정이지만
수업 시간이 절대적으로 부족 !

→ 따라서, 매 수업 시간 개념 설명 후 해당 개념을 실습해 볼 수 있는 문제들을 줄 예정

→ 실습은 필수는 아니나, 모든 개념이 유기적으로 연결되므로 풀어보면 도움이 될 것

 참고 도서

- C언어로 쉽게 풀어 쓴 자료구조 (천인국 저, 한빛미디어)

II. 사전 지식 점검 BY 간단한 문제풀이

두 개의 양의 정수 N과 M을 입력 받아

두 수의 **최대공약수**를 출력하는 프로그램을 for 문을 활용하여 작성하시오.

(* 최대공약수 = Greatest common factor)

입력 예시 1

4 3

출력 예시 1

1

입력 예시 2

16 24

출력 예시 2

8

입력 예시 3

27 36

출력 예시 3

9

II. 사전 지식 점검 BY 간단한 문제풀이

5개의 양의 정수를 입력 받고,
최댓값 (max) 와 최소값 (min) 을 출력하는 프로그램을 작성하시오.

입력 예시 1	출력 예시 1
1 2 3 4 5	max: 5 min: 1
입력 예시 2	출력 예시 2
10 20 12 13 59	max: 59 min: 10
입력 예시 3	출력 예시 3
9 2 3 0 1	max: 9 min: 0

II. 간단한 사전 지식 점검 BY 문제풀이

- 본인 코드를 설명해보자 !

두 개의 양의 정수 N과 M을 입력 받아

두 수의 **최대공약수**를 출력하는 프로그램을 for 문을 활용하여 작성하시오.

(* 최대공약수 = Greatest common factor)

II. 간단한 사전 지식 점검 BY 문제풀이

- 본인 코드를 설명해보자 !

5개의 양의 정수를 입력 받고,
최댓값 (max) 와 최소값 (min) 을 출력하는 프로그램을 작성하시오.

Data Structure란 무엇이고, 왜 필요한가?

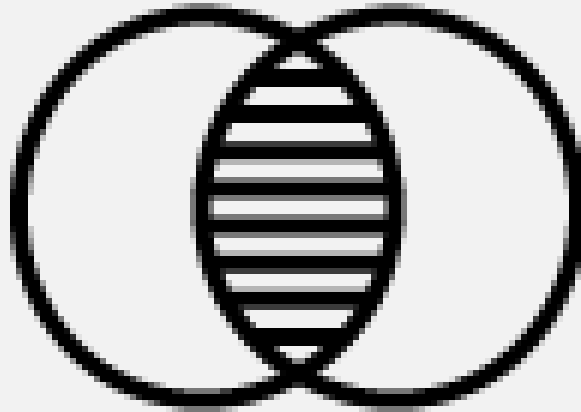
III. Data Structure란 무엇이고, 왜 필요한가?

<https://youtu.be/OH7prOt3vQA>

현실의 문제를 프로그래밍적으로 표현하는 것



TREE



SET



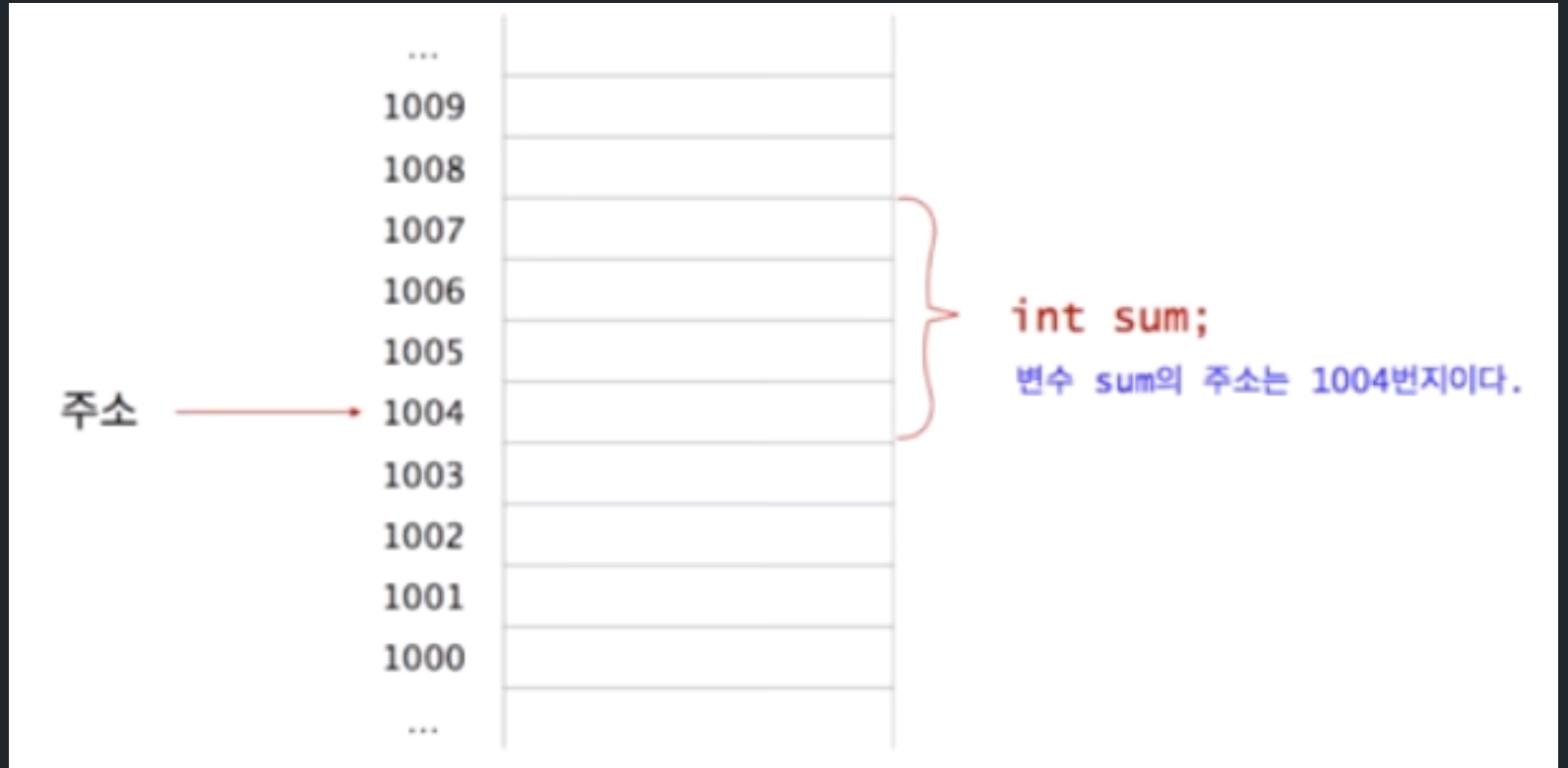
GRAPH

Reviewing C

Pointer / Array / String

I. 메모리 (Memory)

- 컴퓨터의 메모리는 데이터를 보관하는 장소
- 바이트 (8 bits) 단위로 주소가 지정됨
- 모든 변수는 주소를 가진다!



II. 포인터 (Pointer)

- 포인터는 메모리 주소를 값으로 가지는 변수
- 다음과 같이 선언

`type name * variable name;`

- variable name : 선언될 포인터 변수의 이름
- * 는 variable name이 포인터 변수임을 표시
- type name : 포인터 변수에 저장될 주소에 저장될 데이터의 유형을 지정


정수형 포인터 변수

The diagram shows the declaration `int * ptr;` in yellow. Two white arrows point to the components: one from the label '정수형' (Integer type) to the word 'int', and another from the label '포인터 변수' (Pointer variable) to the asterisk '*'.

II. 포인터 (Pointer)

- 연산자 **&**는 변수로부터 그 변수의 **주소를 추출**하는 연산자

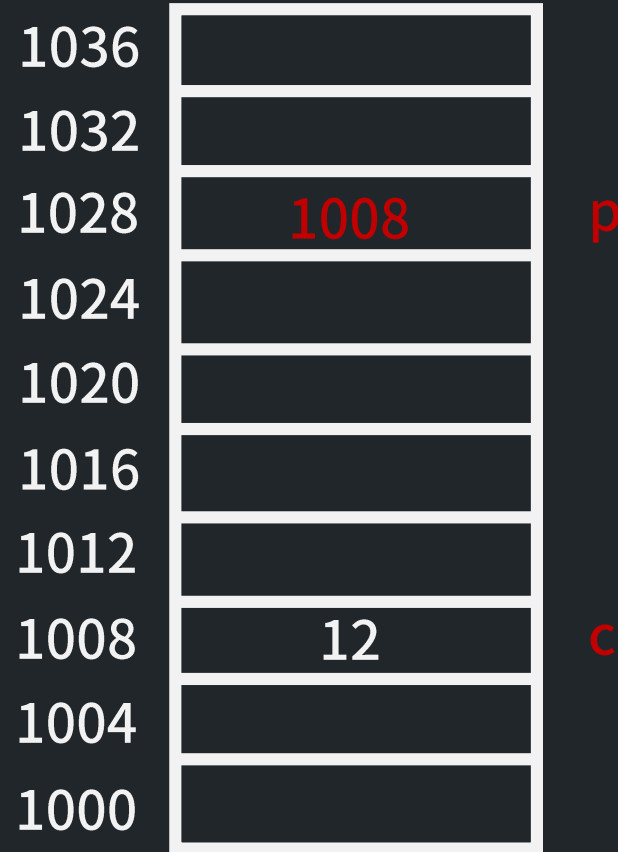
```
int c = 12;
```

```
int *p;
```

```
p = &c;
```



포인터 변수 **p**에 변수 **c**의 주소를 저장한다.



II. 포인터 (Pointer)

```
int x = 1, y = 2;  
int * ip;  
ip = &x;
```

```
int x = 1, y = 2;
```

```
int * ip;
```

```
ip = &x;
```

```
y = * ip;
```

```
* ip = 0;
```

1036		
1032		
1028	1	x
1024	2	y
1020		
1016		
1012		
1008	1028	ip
1004		
1000		

```
y = * ip;
```

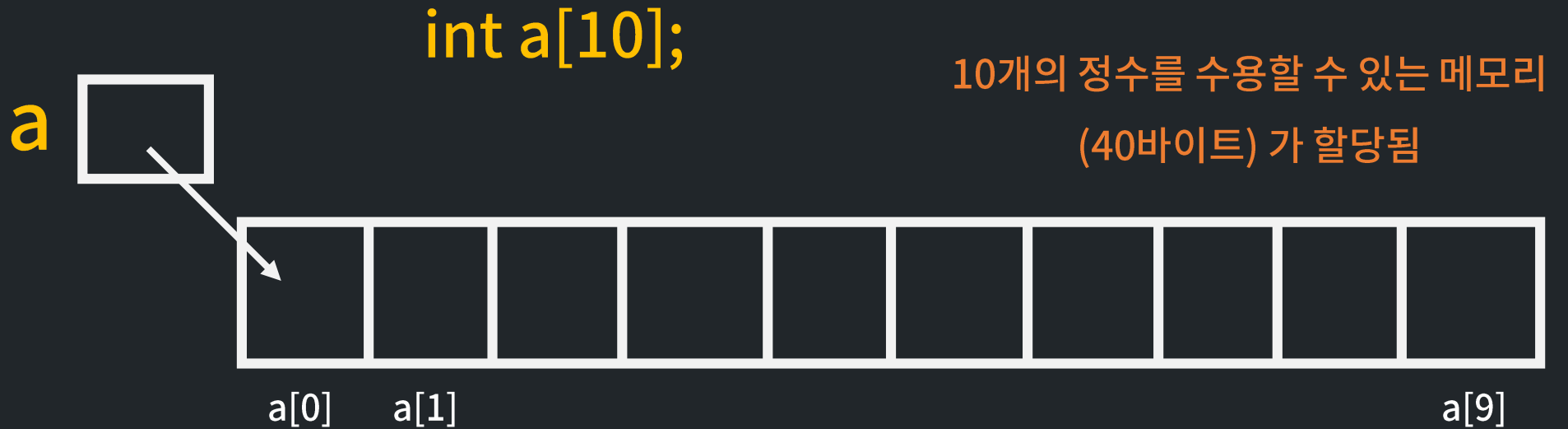
1036		
1032		
1028	1	x
1024	1	y
1020		
1016		
1012		
1008	1028	ip
1004		
1000		

```
* ip = 0;
```

1036		
1032		
1028	0	x
1024	2	y
1020		
1016		
1012		
1008	1028	ip
1004		
1000		

II - 1. 포인터 (Pointer) 와 배열 (Array)

- 포인터와 배열은 매우 긴밀히 연관되어 있음 !
- 예를 들어 다음과 같이 선언된 배열 a가 있다고 하자.



배열의 이름은 배열의 시작 주소를 저장하는 포인터 변수임
(단, 그 값을 변경할 수 없음)

→ 함수의 매개변수로 전달받을 때 포인터 변수를 활용할 수 있다

실습 문제

크기 10인 배열을 선언하고,

자연수 10개를 입력받아 입력받은 자연수들의 총합을 구하는 프로그램을 작성하시오.

조건 1) 합을 계산하는 기능은 메인함수에서 수행하지 말고 **calculate_sum 함수**를 만들어 사용하시오.

조건 2) calculate_sum 함수에서 배열을 전달받을 때, **포인터 변수를 활용**하시오.

입력예시 : 1 2 3 4 5 6 7 8 9 10

출력예시 : 55

Thank You