

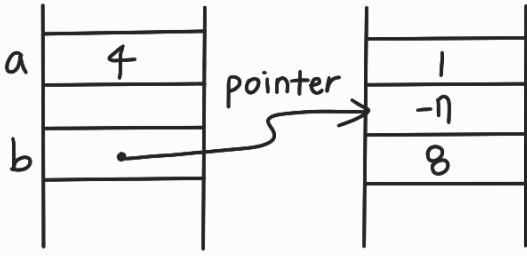
## Ch 3 배열, 구조체, 포인터

# 1. 배열

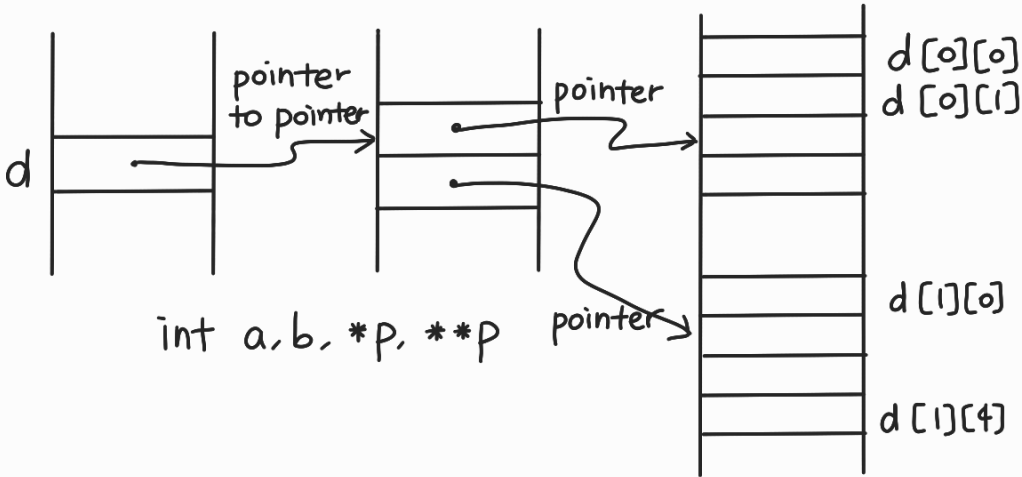
```
int a, b[3], d[2][5];
```

5개씩 2개

- ( ① homogeneous (동질적)
- ② Continuous (연속적)



### 사용 예

$$a = 4$$
$$b[0] = 1 \quad b[1] = -1$$
$$b[b[0]+1] = b[1] + 15$$


memory 할당 (allocation)

### ① 정적 할당 (static memory allocation)

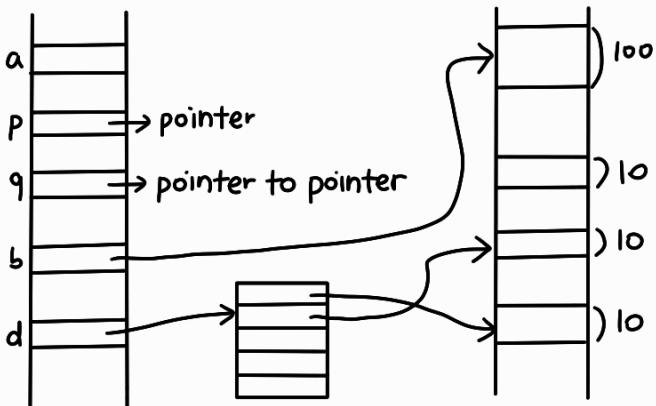
: 선언문에 의해 시작된 고정 할당

## ② 동적 할당 (Dynamic memory allocation)

: 실행중에 요청에 의해 임시적 할당 및 제거

malloc(size) free()

```
int a, *p, **q, b[100], d[5][10];
```



$p=b$  // 공간 공유

$$p[2] = 4 \Leftrightarrow b[2] = 4 \quad // \text{사용}$$

or `p = (int*) malloc (100 * sizeof(int))`

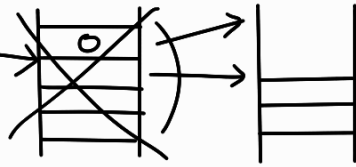
// 동전할당

$p[2] = 4$  // 사용

free(p) // 반납

q = d

or `q = (int**) malloc (5 * sizeof (int*))`



`q[0] = (int *) malloc (10 * sizeof (int))`

`q[1] =` "

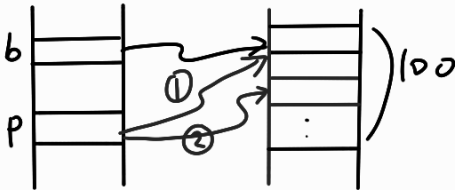
`...`

`q[4] =` "

`q[0][2] = ...`

`{ free(q[0]), free(q[1]), ... free(q[4])`  
`free(q) ← 공간 할당의 역순으로 반납`

✗ `int b[100], *p;`



① `p = b` ← p가 b를 공유

`p[4] = -1 ⇔ b[4] = -1`

② `p = b + 2` ← p가 b[2]에서 시작하는 배열 공유

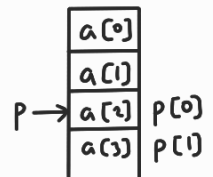
`p[4] = -1 ⇔ b[2] = -1`

`int a[10], *p, **q;`

`for (int i = 0; i < 10; i++) a[i] = i;`

`p = &a[2];`

`cout << p[1]; // 3`



## 2. 구조체 (structure)

: 사용자가 정의하는 새로운 data type

서로 다른 data type의 원소들을 모아서 하나의 record로 구성

- ① heterogeneous (이질적)
- ② continuous (연속적)

ex) `typedef struct`

`{`

`char name[20];` → 이름 문자열 (string)

`int age;` → 나이

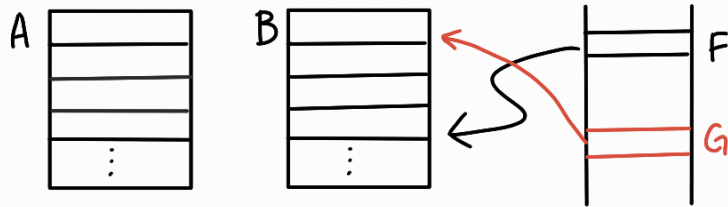
`float height;` → 키

`char add[100];` → 주소

`...`

`} student;` → 새로운 datatype의 이름

student A, B, D[10], E[3][5], \*F, \*G;



$F = (\text{student}^*) \text{ malloc } (10 * \text{sizeof}(\text{student}))$

~~free(F)~~

사용법

$G = \&B$

$A.name = "BP"$

$B.name = "BP"$

구조체 그 자체

구조체를 가리키는 포인터

$G \rightarrow name = "BP" \Leftrightarrow (*G).name = "BP"$

$G \rightarrow age = 19 \Leftrightarrow (*G).age = 19$

✖ ✖  $G[4].name = "BP"$

type struct

{ char name[20];

int age;

float height;

student A, B;

...

} Node; → 새로운 구조체의 이름

type struct

{ char name[20];

int age;

float height;

~~Node A, B;~~ 불가능. Node 미완성

...

} Node;

type struct

{ char name[20];

int age;

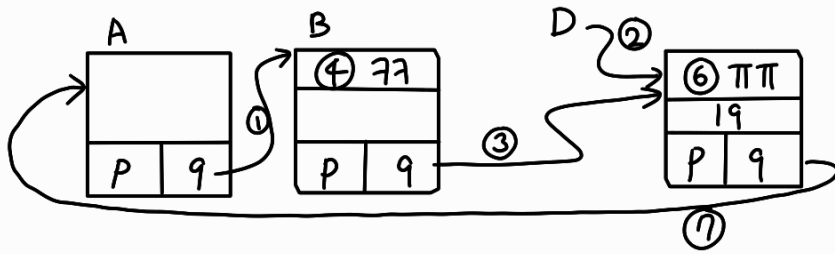
float height;

Node \*p, \*q; (가능) → 자체 pointer (Link)

...

} Node;

Node A, B, \*D



①  $A \cdot q = \&B$

②  $D = (\text{Node}^*)\text{malloc}(\text{sizeof}(\text{Node}))$

③  $B \cdot q = D$  주소를 담고 있음

④  $A \cdot q \rightarrow \text{name} = "77"$

⑤  $D \rightarrow \text{age} = 19$  ( $\Leftrightarrow B \cdot q \rightarrow \text{age} = 19$ )

⑥  $B \cdot q \rightarrow \text{name} = "ππ"$

⑦  $D \rightarrow q = \&A$