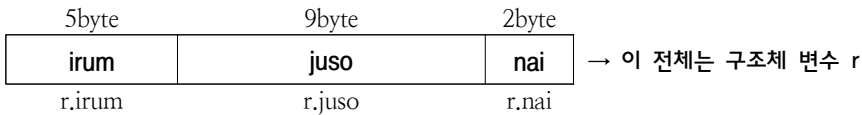


## 9. 구조체(struct)

- ① 구조체는 여러 개의 변수를 하나의 자료형으로 묶어서 취급한다.(레코드 구조)
- ② 구조체는 구조체를 구성하는 멤버 단위로 취급할 수도 있고, 구조체 변수를 이용하여 구조체를 하나의 단위로 취급할 수 있다.
- ③ 구조체 멤버를 개별적으로 다룰 때는 **Dot 연산자(.)**를 이용한다.

```

└───┬─── 구조체 태그(Tag)
struct insa{
    char  irum[5];  └
    char  juso[9];  | 구조체 멤버
    short nai;     └
} r; ──> 구조체 변수(insa형 구조체 변수 r)
    
```



◆ 구조체를 다루는 기본 프로그램

```

void main()
{
    struct rec{
        char ch;
        int k;
    }r1;
    struct rec r2;    //새로운 구조체 변수 r2를 선언
    r1.ch = 'A';      //구조체 멤버 ch에 'A'를 대입
    r1.k = 30;        //구조체 멤버 k에 30을 대입
    r2 = r1;          //구조체 변수 r2에 r1을 한꺼번에 대입
    printf("%c\n", r2.ch);
    printf("%d\n", r2.k);
}
    
```

실행결과 : A

◆ 구조체 배열과 포인터

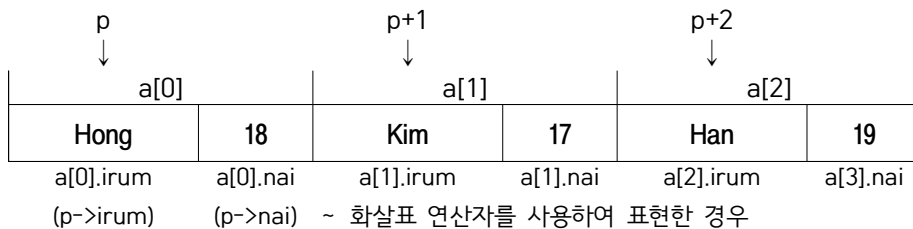
- 구조체 변수에 배열과 포인터가 사용되는 경우이다.
- 구조체 변수가 포인터이면 **화살표 연산자(->)**를 이용하여 멤버를 취급한다.
- p가 구조체 포인터이면 다음 두 문장은 동등하다.

(\*p).irum ~ dot 연산자를 사용하는 경우  
||  
p->irum ~ arrow 연산자를 사용하는 경우

```
void main()
{
    struct insa{
        char irum[10];
        int nai;
    }a[] = {"Hong", 18, "Kim", 17, "Han", 19}; //구조체 배열
    struct insa *p; //구조체 포인터 p
    p = a;
    p++; //구조체 포인터 p 증가
    printf("%s\n", p->irum);
    printf("%d\n", p->nai);
}
```

실행결과 : Kim  
17

위 프로그램의 메모리 구조는 다음과 같다.



**//탐구 -----**  
**typedef**

typedef은 사용자가 새로운 자료형을 정의할 때 사용한다.

**[형식]** typedef 기존자료형 새로운자료형 ;

- typedef char byte; → ‘char’ 대신에 'byte'를 자료형으로 사용 가능
- typedef char \* string; → ‘string’이라는 새로운 char형 포인터 자료형을 정의
- typedef int array[80]; → ‘array’이라는 새로운 int형 배열 자료형을 정의

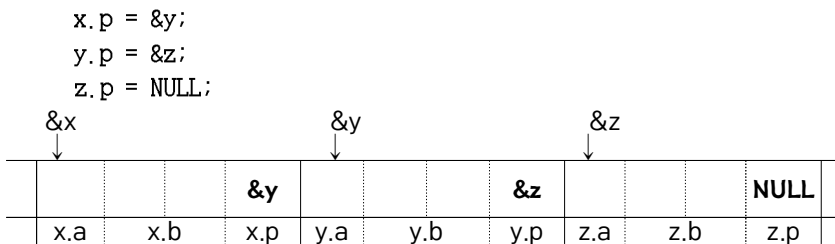
**[Tip]** typedef에서 typedef를 없애고 보면, 새로운 자료형은 변수 위치에 있다.

◆ C에서 자기참조 구조체

C에서 자기참조 구조체를 선언 방법이다. 다음 3가지는 같은 뜻이다.

[방법1]	[방법2]	[방법3]
<pre>struct rec{     char a;     short b;     struct rec *p; } x, y, z;</pre>	<pre>typedef struct rec{     char a;     short b;     struct rec *p; } node; node x, y, z;</pre>	<pre>typedef struct rec rec; struct rec {     char a;     short b;     rec *p; }; rec x, y, z;</pre>

위의 방법 중 어느 한가지로 선언하고, 다음처럼 기술하면 메모리 구조는 다음과 같다.



- 구조체 변수 x의 멤버인 x.p는 자기와 똑같은 구조를 갖는 변수 y의 주소를 가리킨다.

기출문제 분석

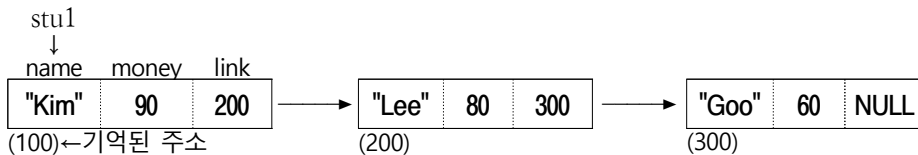
1. 다음 C 프로그램의 실행 결과는? [2016년 서울 9급]

```
#include<stdio.h>
struct student
{
    char name[20];    // 이름
    int money;        // 돈
    struct student *link; // 자기 참조 구조체 포인터 변수
};
int main(void)
{
    struct student stu1 = {"Kim", 90, NULL};
    struct student stu2 = {"Lee", 80, NULL};
    struct student stu3 = {"Goo", 60, NULL};
    stu1.link = &stu2;
    stu2.link = &stu3;
    printf("%s %d\n", stu1.link->link->name, stu1.link->money);
    return 0;
}
```

- ① Goo 80                      ② Lee 60                      ③ Goo 60                      ④ Lee 80

☞ 자기 참조 구조체

// 구조체 포인터 link는 자신과 똑같은 구조를 갖는 구조체의 주소를 가리킨다.



- stu1.name = "Kim"
- stu1.money = 90
- stu1.link = 200
- printf("%s %d\n", stu1.link->link->name, stu1.link->money);
- stu1.link->link->name = "Goo"
- stu1.link->money = 80