

## 10. 매개변수 전달

매개변수 전달 방식은 크게 5가지가 있다.

현재, 시험에 집중적으로 출제되는 것은 다음 2가지이다.

### ◆ call by value

- 실매개변수의 값이 대응하는 형식매개변수에 일대일로 전달되는 기법이다.
- 형식매개변수를 위한 별도의 기억장소가 필요하다.
- 실매개변수와 형식매개변수는 서로 다른 기억장소를 각각 갖게 된다.
- 형식매개변수의 값이 변해도 실매개변수의 값은 변하지 않는다.
- 함수가 종료되면, 형식매개변수의 기억공간은 소멸된다.

### ◆ call by reference

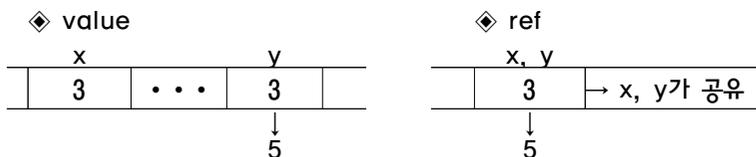
- 실매개변수 주소가 형식매개변수에 전달되어 같은 기억장소를 서로 공유하는 방식이다.
- 형식매개변수를 위한 별도의 기억장소가 필요없다.
- 따라서, 형식매개변수의 값이 변하면 실매개변수의 값도 변하게 된다.

[예제] 매개변수 전달방식이 call by value와 call by reference일 때 각각 출력되는 값은?

```
void foo(int y){ y = y + 2; } //y는 형식매개변수(가인수)
void main()
{
    int x = 3;
    foo(x);           //x는 실매개변수(실인수)
    printf("%d\n", x);
}
```

- ① 3, 3                      ② 5, 3  
 ③ 5, 5                      ④ 3, 5

정답 : ④



기출문제 분석

1. 다음 C 프로그램의 실행 결과는? [2017년 서울 9급]

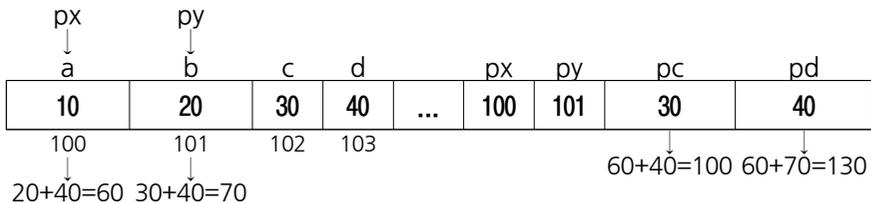
```
#include <stdio.h>
void change(int *px, int *py, int pc, int pd);
void main(void)
{
    int a=10, b=20, c=30, d=40;
    change(&a, &b, c, d);
    printf("a=%d b=%d c=%d d=%d", a, b, c, d);
}
void change(int *px, int *py, int pc, int pd)
{
    *px=*py+pd; *py=pc+pd;
    pc=*px+pd; pd=*px*py;
}
```

- ① a=60 b=70 c=50 d=30
- ② a=60 b=70 c=30 d=40
- ③ a=10 b=20 c=50 d=30
- ④ a=10 b=20 c=30 d=40

☞ 인수전달

• `change(&a, &b, c, d);`  
 ↓                      ↘ 내용 전달(r-value 전달)  
 주소 전달(l-value 전달)

• 메모리 구조는 다음과 같다.



2. 다음 C 프로그램의 출력 결과는? [2018년 지방 9급]

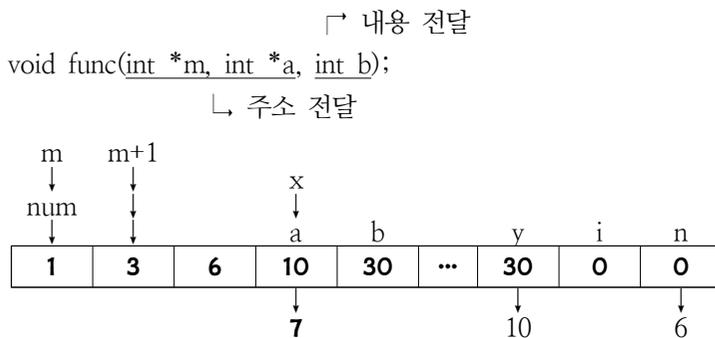
```

-----
#include <stdio.h>
#define SIZE 3
void func(int *m, int *a, int b);
int main(void)
{
    int num[SIZE] = { 1, 3, 6 };
    int a=10, b=30;
    func(num, &a, b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
void func(int *m, int *x, int y)
{
    int i = 0, n = 0;
    y = *x;
    n = *(m + 1) + (*m + 2);
    *x = ++n;
}
-----
    
```

- ① a = 7, b = 10    ② a = 7, b = 30  
 ③ a = 10, b = 10    ④ a = 10, b = 30

♣ 매개변수 전달

• 메모리 구조



- $n = *(m + 1) + (*m + 2) = 3 + (1 + 2) = 6$
- `printf("a = %d, b = %d\n", a, b);` // a = 7, b = 30

3. C 언어에서 함수 호출 시 매개변수 전달 방법에는 값에 의한 호출(call by value)과 참조에 의한 호출(call by reference)이 있다. C 프로그램 코드가 다음과 같을 때 설명으로 옳지 않은 것은? [2022년 국가 9급]

```
-----  
int get_average(int score[], int n) {  
    int i, sum;  
    for(i = 0; i < n; i++) sum += score[i];  
    return sum / n;  
}  
void main(void) {  
    int score[3] = { 1, 2, 5 };  
    printf("%d\n", get_average(score, 3));  
}  
-----
```

- ① 전달할 데이터의 양이 많을 경우에는 참조에 의한 호출이 효율적이다.
- ② 값에 의한 호출로 전달된 데이터는 호출된 함수에서 값을 변경하더라도 함수 종료 후 해당 함수를 호출한 상위 함수에 반영되지 않는다.
- ③ 값에 의한 호출은 함수 호출 시 데이터 복사가 발생한다.
- ④ 위의 프로그램에서 함수 get\_average()를 호출하는 데 사용한 매개변수 score는 값에 의한 호출로 처리된다.

♣ 매개변수 전달

```
-----  
int get_average(int score[], int n) {  
    int i, sum = 0; // 주어진 문제에서 sum은 0으로 초기화되어 있지 않음  
    for(i = 0; i < n; i++) sum += score[i];  
    return sum / n;  
}  
void main(void) {  
    int score[3] = { 1, 2, 5 };  
    printf("%d\n", get_average(score, 3)); // 출력 = (1 + 2 + 5) / 3 = 2.666667 = 2  
}                                     ↓   ↘ 값 전달  
                                     주소 전달
```

- C의 매개변수 전달방식은 기본적으로 값에 의한 호출(call by value)이다.
- C에서 포인터를 매개변수로 전달하면 참조에 의한 호출(call by reference) 효과가 있게 된다.

4. 다음의 C 프로그램 실행 결과로 출력되는 a, b, c 값으로 옳은 것은? [2018년 국회 9급]

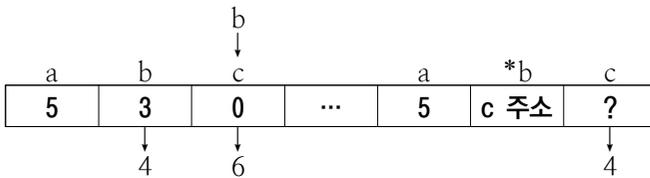
```

-----
#include <stdio.h>
int foo(int a, int *b) {
    int c;
    *b = a + 1;
    c = a - 1;
    return c;
}
void main() {
    int a = 5;
    int b = 3;
    int c = 0;
    b = foo(a, &c);
    c = foo(b, &a);
    printf("a=%d b=%d c=%d\n", a, b, c);
}
-----
    
```

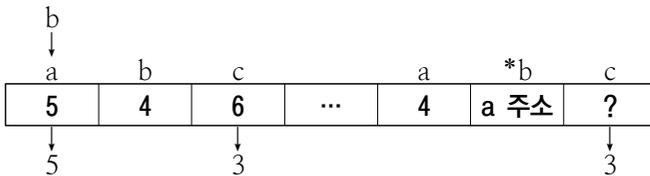
- ① 5, 4, 3                      ② 4, 3, 2                      ③ 6, 3, 4
- ④ 4, 3, 5                      ⑤ 5, 3, 0

☞ 인수 전달

• b = foo(a, &c); //a는 값 전달, c는 c의 주소 전달



• c = foo(b, &a); //b는 값 전달, a는 a의 주소 전달



a = 5, b = 4, c = 3

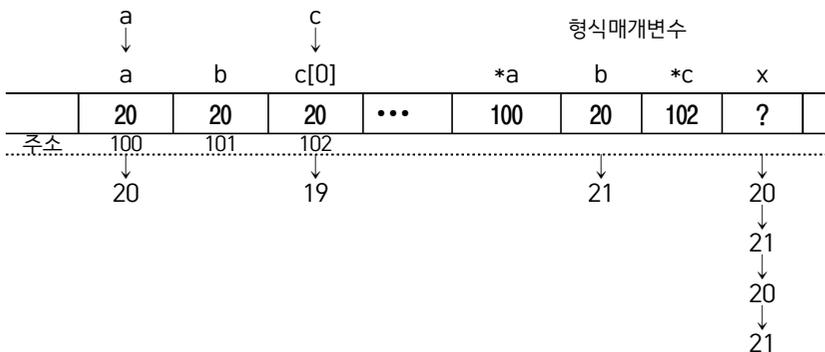
5. 다음 C 프로그램의 출력 결과로 옳은 것은? [2014년 국가 9급]

```
#include<stdio.h>
void func(int *a, int b, int *c){
    int x;
    x = *a;
    *a = x++;
    x = b;
    b = ++x;
    --(*c);
}
int main(){
    int a, b, c[1];
    a = 20;
    b = 20;
    c[0] = 20;
    func(&a, b, c);
    printf("a = %d b = %d c = %d\n", a, b, *c);
    return 0;
}
```

- ① a = 20 b = 20 c = 19
- ② a = 20 b = 21 c = 19
- ③ a = 21 b = 20 c = 19
- ④ a = 21 b = 21 c = 20

☞ 매개변수 전달

• 메모리 구조 값 변화는 다음과 같다.



- 함수 func()가 종료되면 형식매개변수 영역은 소멸된다.
- 출력 결과 : a = 20, b = 20, \*c = 19

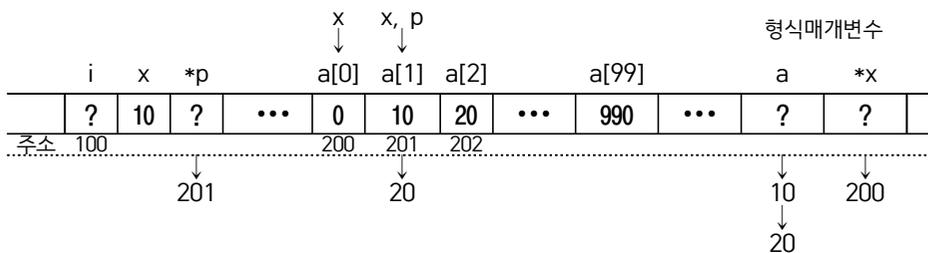
6. 다음 C 프로그램의 출력 결과로 옳은 것은? [2014년 국회 9급]

```
#include <stdio.h>
int *func(int a, int *x){
    a = a + 10;
    x = x + 1;
    *x = *x * 2;
    return x;
}
int main( )
{
    int i;
    int x = 10;
    int *p;
    int a[100];
    for(i = 0; i < 100; i++) a[i] = i * 10;
    p = func(x, a);
    printf("sum = %d\n", x + a[0] + a[1] + p[0] + p[1]);
}
```

- ① 오류 발생                      ② 60                                  ③ 61
- ④ 70                                  ⑤ 80

☞ 배열과 포인터 / 매개변수 전달 - 복잡한 문제(?)

• 메모리 구조는 다음과 같으며, 값은 다음처럼 변한다.



$$\text{수식 합} = x + a[0] + a[1] + p[0] + p[1] = 10 + 0 + 20 + 20 + 20 = 70$$

정답 : ④