7장 메모리 주소 관리하기 : 포인터

1. 이 장에서 만드는 프로그램
2. 포인터란
3. 포인터로 배열 다루기
4. 프로젝트: 로또번호 생성

1. 이 장에서 만드는 프로그램

1. 배열 안의 값을 오름차순 정렬 하시오

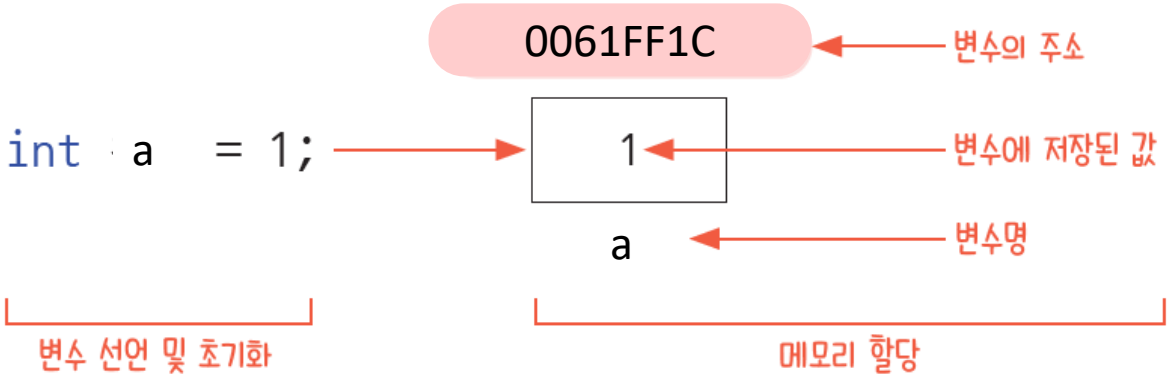
2. 중복되지 않은 로또 번호 6개를 생성하고 오름차순 정렬하시오

2. 포인터란

(1) 변수와 메모리의 관계

- 철수 변수 선언 및 초기화 → 메모리에서 0061FF1C라는 위치에 철수라는 이름의 공간을 할당하고 그 안에 1이라는 값을 넣는다.

그림 7-3 변수와 메모리의 관계



2. 포인터란

(1) 변수와 메모리의 관계

```c
int main(void){
    int a = 1;
    int b = 2;
    int c = 2;
    // 변수의 주소와 변수값 출력
    printf("a변수 주소 : %p, a변수 값 : %d\n", &a, a);
    printf("b변수 주소 : %p, b변수 값 : %d\n", &b, b);
    printf("c변수 주소 : %p, c변수 값 : %d\n", &c, c);
    return 0;
}
```

실행결과

```
a변수 주소 : 0061FF1C, a변수 값 : 1
b변수 주소 : 0061FF18, b변수 값 : 2
c변수 주소 : 0061FF14, c변수 값 : 2
```

2. 포인터란

(2) 포인터로 다른 변수의 주소와 값 알아내기

- 포인터 변수 또는 포인터 : *를 넣어 선언한 변수

형식 자료형 * 변수명;

- pointer 변수의 주소값 : 각 변수의 주소값 할당
- Pointer 주소는 0061FF1C이고 그곳의 값은 1

2. 포인터란

(2) 포인터로 다른 변수의 주소와 값 알아내기

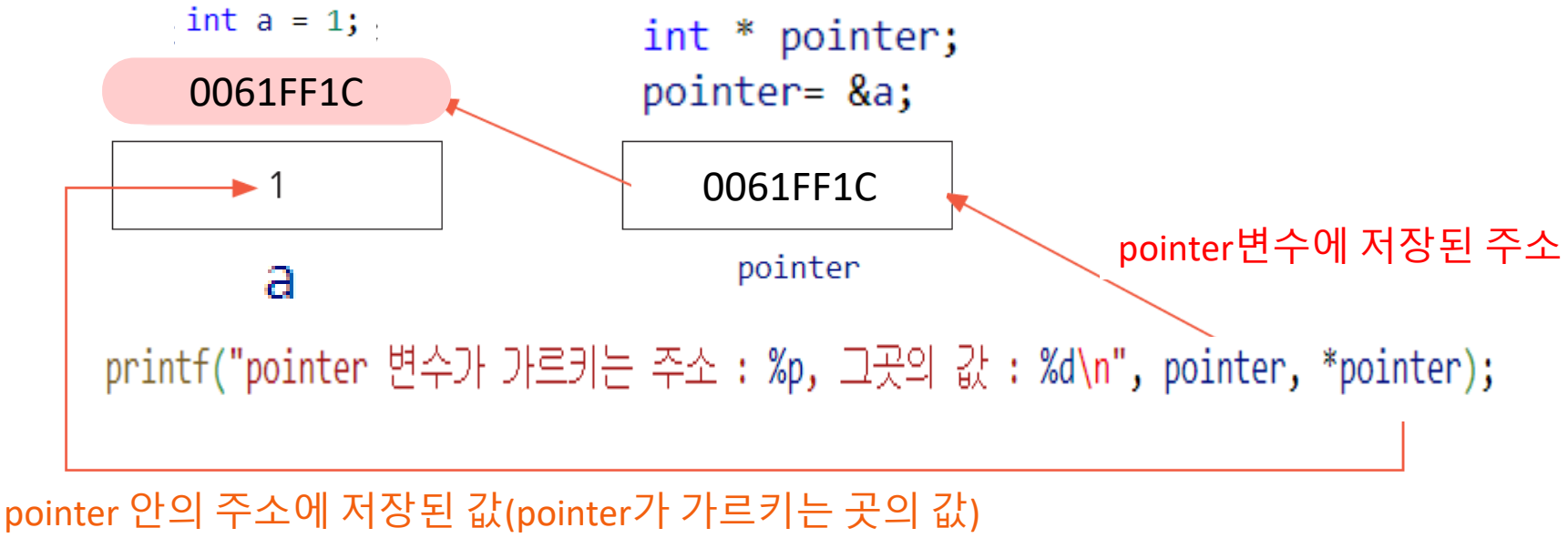
```
실행결과
a변수 주소 : 0061FF18, a변수 값 : 1
b변수 주소 : 0061FF14, b변수 값 : 2
c변수 주소 : 0061FF10, c변수 값 : 2

pointer 변수가 가르키는 주소 : 0061FF18, 그곳의 값 : 1
pointer 변수가 가르키는 주소 : 0061FF14, 그곳의 값 : 2
pointer 변수가 가르키는 주소 : 0061FF10, 그곳의 값 : 2
```

```c
int main(void){
    int a = 1;
    int b = 2;
    int c = 2;
    // 변수의 주소와 변수값 출력
    printf("a변수 주소 : %p, a변수 값 : %d\n", &a, a);
    printf("b변수 주소 : %p, b변수 값 : %d\n", &b, b);
    printf("c변수 주소 : %p, c변수 값 : %d\n\n", &c, c);
    // 포인터 변수 선언과 값 출력
    int * pointer = &a;
    printf("pointer 변수가 가르키는 주소 : %p, 그곳의 값 : %d\n", pointer, *pointer);

    pointer = &b;
    printf("pointer 변수가 가르키는 주소 : %p, 그곳의 값 : %d\n", pointer, *pointer);

    pointer = &c;
    printf("pointer 변수가 가르키는 주소 : %p, 그곳의 값 : %d\n", pointer, *pointer);
    return 0;
}
```

2. 포인터란

(2) 포인터로 다른 변수의 주소와 값 알아내기

<그림> 변수와 포인터 변수의 관계



2. 포인터란

(3) 포인터로 다른 변수의 값 바꾸기

- pointer가 가르키는 곳의 값에 3을 곱하기
- *pointer에 곱하기 3을 하고 이를 다시 *pointer에 저장
- 포인터 변수는 다른 변수의 주소를 알아낼 수 있고, 알아낸 변수의 주소에 찾아가 값도 직접 바꿀 수 있다.

```
// 포인터 변수 선언과 값 출력
int * pointer;
pointer= & a ;
*pointer = *pointer * 3;
printf("pointer 변수가 가르키는 주소 : %p, 그곳의 바뀐값 : %d\n", pointer, *pointer);
```

2. 포인터란

(4) 포인터 추가하기

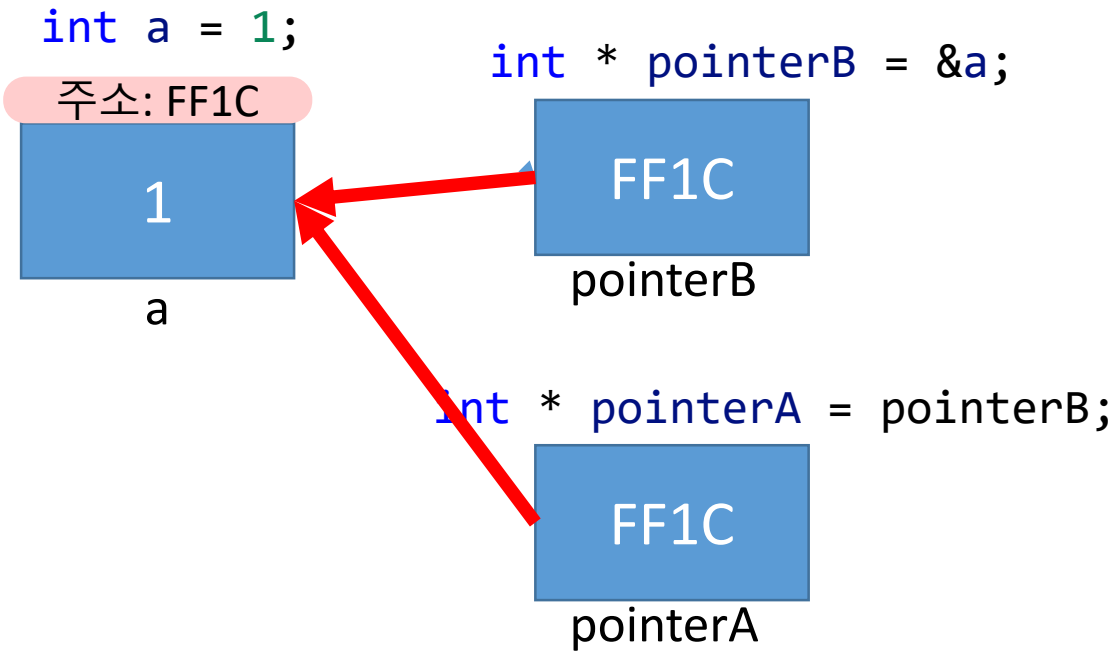
- pointerB : pointerA이 모르게 pointerA가 가르키는 곳의 값을 수정
- 포인터 변수 2개는 같은 주소, 즉 하나의 메모리 공간을 가리킬 수 있다.
- 포인터 변수로 주소를 알고 있으면 해당 주소에 가서 저장된 값을 가져올(읽어 올) 수도 있고 값을 변경할 수도 있다.

```
int * pointerA = pointerB;
```

2. 포인터란

(4) 포인터 추가하기

<그림> 동일한 주소를 가리키는 포인터 변수



2. 포인터란

(4) 포인터 추가하기

- 포인터 변수는 어떤 변수의 주소를 값으로 가지고 있으며 주소를 이용해 주소에 해당하는 변수의 값을 직접 바꿀 수 있다.
- 포인터 변수도 다른 변수와 마찬가지로 주소가 있다.

1분 퀴즈

1. 다음 중 포인터 변수를 잘 이해하고 있는 친구를 모두 고르세요

[보기]

- 선영 : 포인터 변수는 메모리의 주소값을 저장해
- 지호 : 포인터 변수는 int * p; 처럼 *를 이용해 선언하지
- 준현 : 변수의 메모리 주소는 변수명 앞에 &를 붙이면 확인할 수 있어

1분 퀴즈

2. 다음 코드를 실행했을 때 출력되는 내용을 고르세요

```
int a = 10;
int * p = &a;
a = 15;
printf("%d\n", *p);
```

①10　　②15

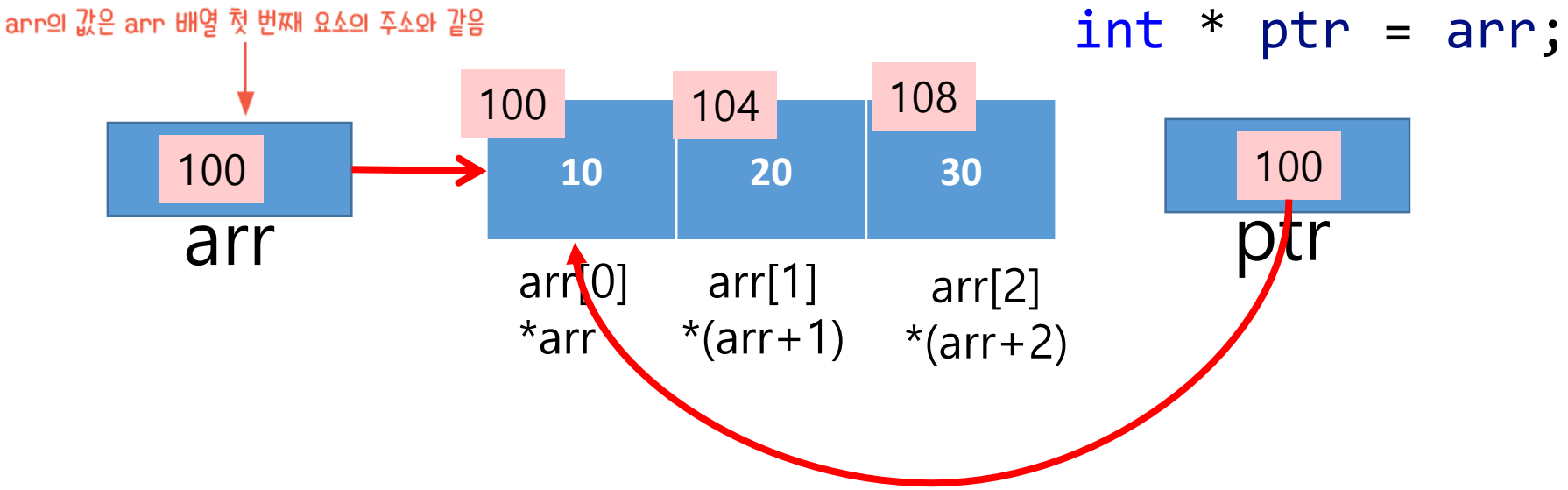
③a변수의 메모리주소　　④오류발생

3. 포인터로 배열 다루기

(1) 포인터로 배열에 접근하기

<그림>포인터 변수와 배열의 관계

```
int arr[3] = { 5, 10, 15 };
```



3. 포인터로 배열 다루기

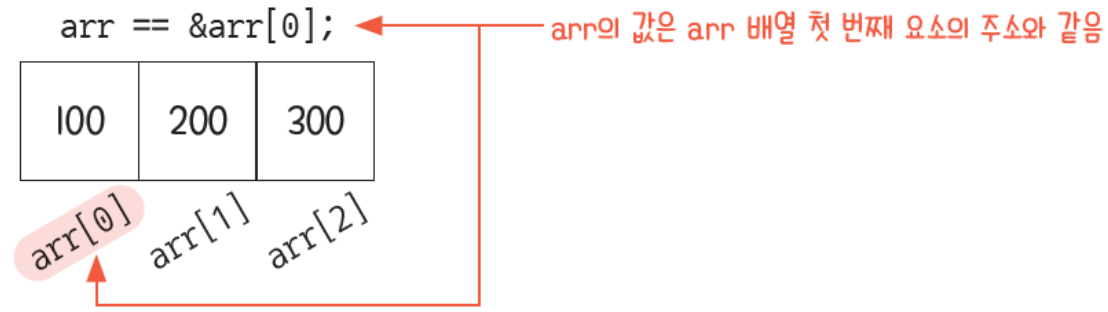
(1) 포인터로 배열에 접근하기

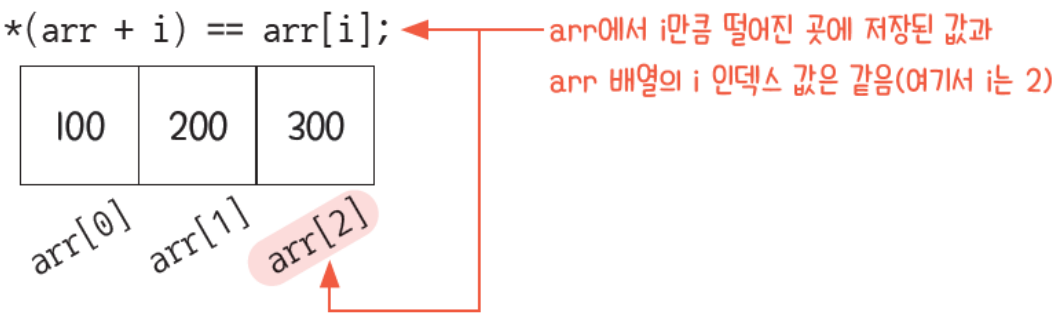
```
int main(void){
    int arr[3] = { 5, 10, 15 };
    for(int idx=0 ; idx<3 ; idx++){
        printf("배열 arr[%d] = %d\n", idx, arr[idx]);
    }
    int * ptr = arr;
    for(int idx=0 ; idx<3 ; idx++){
        printf("포인터변수 ptr[%d] = %d\n", idx, ptr[idx]);
    }
    ptr[0] = 100; ptr[1] = 200; ptr[2] = 300;
    printf("\n === 수정후 === \n");
    for(int idx=0 ; idx<3 ; idx++){
        printf("배열 arr[%d] = %d\n", idx, arr[idx]);
        printf("배열 arr[%d] = %d\n", idx, *(arr+idx));
    }
    for(int idx=0 ; idx<3 ; idx++){
        printf("포인터변수 ptr[%d] = %d\n", idx, ptr[idx]);
        printf("포인터변수 ptr[%d] = %d\n", idx, *(ptr+idx));
    }
}
```

3. 포인터로 배열 다루기

(1) 포인터로 배열에 접근하기

그림 7-7 배열명은 배열 첫 번째 요소의 주소(시작 주소)를 나타냄





3. 포인터로 배열 다루기

(2) 실습 1: 포인터로 두 변수의 값 교환하기

두 변수의 값을 교환하는 방법

㉠ 두 변수를 전달받아 두 변수의 값을 교환하는 swap() 함수 선언

㉡ 변수 a, b 선언, 초깃값으로 각각 10, 20

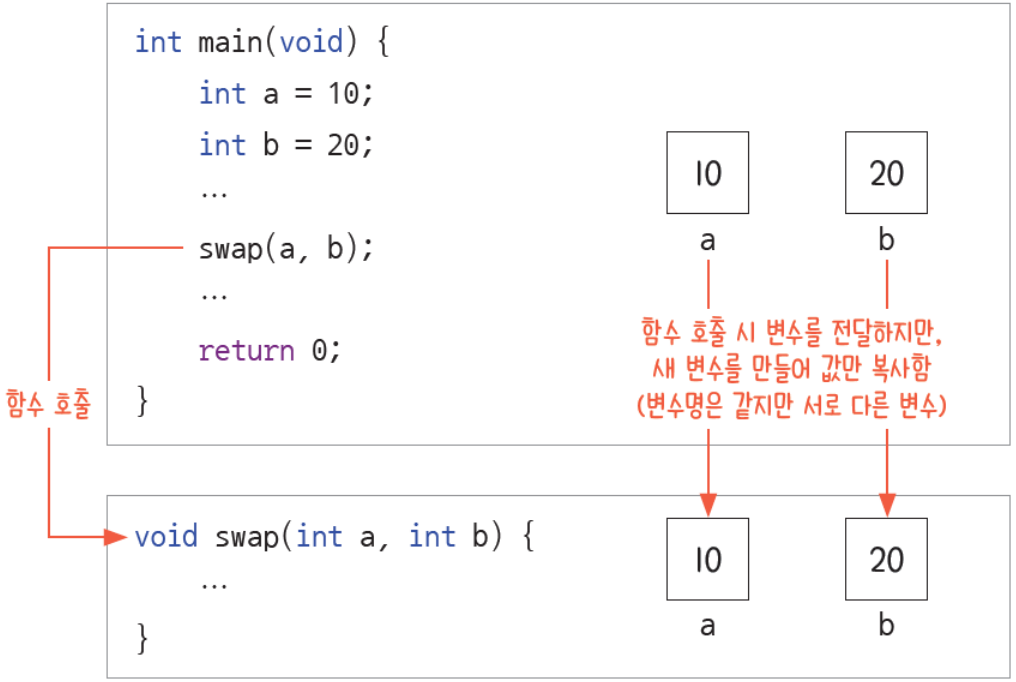
㉢ main() 함수 뒤에 swap() 함수 정의, temp 변수 선언

④ main() 함수에서 swap() 함수 호출

3. 포인터로 배열 다루기

(2) 실습 1: 포인터로 두 변수의 값 교환하기

- 값에 의한 호출(call by value) : 함수를 호출하면서 전달값으로 변수를 넘기면 호출한 함수 안에서는 변수 자체가 아닌 전달받은 변수의 값만 복사해서 ...



3. 포인터로 배열 다루기

(2) 실습 1: 포인터로 두 변수의 값 교환하기

변수의 주소를 전달받는 swap_addr() 함수를 추가하기

㉠ main() 함수 앞에 swap_addr() 함수 선언,
매개변수 a와 b 앞에 * 추가

㉡ main() 함수 뒤에 swap_addr() 함수 정의

㉢ temp 변수에는 주소가 아닌 실제 값을 저장,
포인터 변수 a에 담긴 주소의 실제 값을 temp 변수에 저장

④ 두 변수 앞에 * 추가

⑤ b 앞에만 * 추가

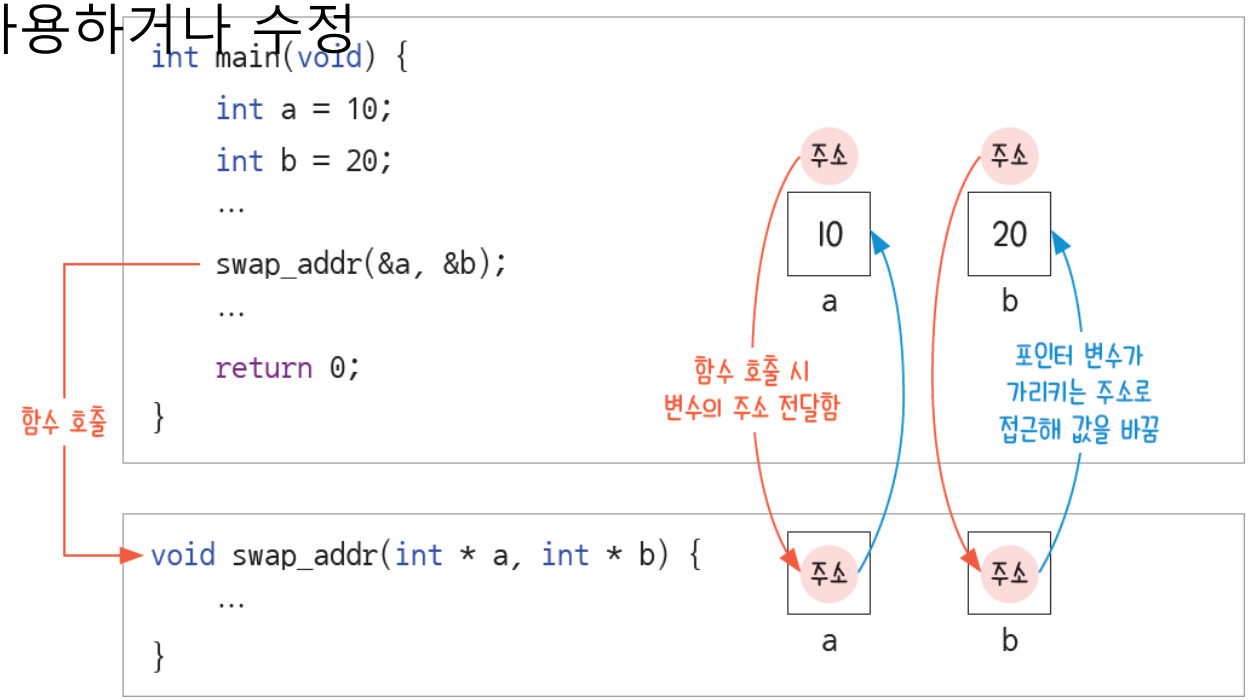
⑥ a와 b 앞에 * 추가

⑦ 변수 앞에 & 추가

3. 포인터로 배열 다루기

(2) 실습 1: 포인터로 두 변수의 값 교환하기

- 참조에 의한 호출(call by reference) : 함수를 호출하면서 전달 값으로 변수의 주소를 넘기면 호출한 함수 안에서 변수의 주소를 참조해 값을 사용하거나 수정



3. 포인터로 배열 다루기

(3) 실습 2: 포인터로 배열의 값 바꾸기

㉠ main() 함수 앞에 changeArray() 함수 선언
매개변수를 포인터 변수 ptr로 선언

㉡ main() 함수에 크기가 3인 arr2라는 이름의 배열을 선언하고 값을 넣어 초기화

㉢ main() 함수 뒤에 changeArray() 함수 선언을 가져와 정의
배열의 세 번째 요소는 인덱스로 2이므로 ptr[2]로 표시하고 여기에 50을 넣기

④ main() 함수에서 changeArray() 함수 호출
&를 붙일 필요 없이 arr2를 그대로 changeArray() 함수에 전달

⑤ for 문을 사용해 배열의 값 출력

3. 포인터로 배열 다루기

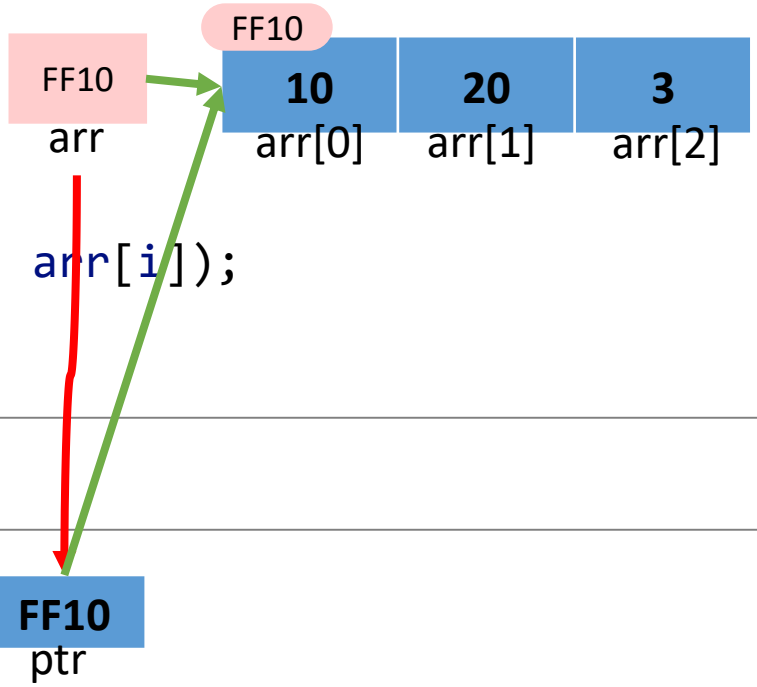
(3) 실습 2: 포인터로 배열의 값 바꾸기

<그림>포인터로 배열의 값 바꾸기

```
int main(void){
    int arr[3] = {10, 20, 3};
    changeArray(arr);
    for(int i=0 ; i<3 ; i++){
        printf("arr[%d] = %d\n", i, arr[i]);
    }
}
```

함수호출

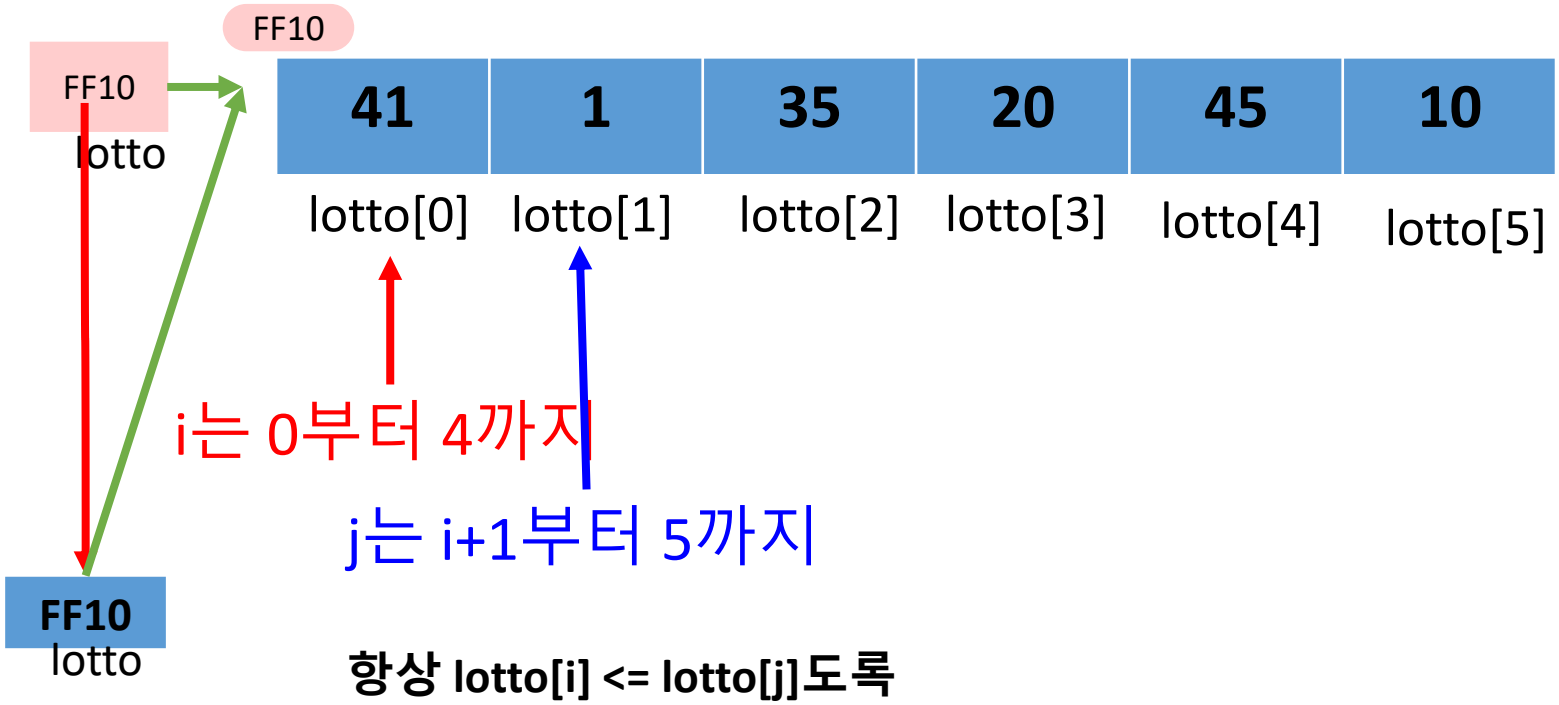
```
void changeArray(int * ptr){
    ptr[1] = 99;
}
```



4. 프로젝트1: 미니 정렬 프로그램

배열 안의 값을 오름차순 정렬하는 프로그램을 작성하시오

(void sort(int* arr, int cnt)함수 작성 : 배열의 길이가 cnt인 arr배열을 정렬하는 함수작성한 후 main에서 호출)



4. 프로젝트2: 로또 번호 생성기

중복되지 않는 로또 번호 6개를 생성한 후, 오름차순 정렬하는 프로그램을 작성하시오

- void sort(int* lotto)함수 작성 : lotto 배열을 정렬하는 함수작성한 후 main에서 호출
- void make_lotto(int * lotto) 또는 int* make_lotto()

```
int* make_lotto(){
    static int lotto[6];
    srand((unsigned int) time(NULL)); // 난수 초기화
    // for(int i=0 ; i<6 ; i++){
    //     lotto[i] = rand()%45 +1;
    // }
    int i = 0;
    while(i<6){
        int temp = rand()%45 + 1;
        int duplication_check = 1;
        // 뽑은 난수 temp가 이전에 뽑은 난수와 중복되는지 체크 부분

        if(duplication_check){
            lotto[i] = temp;
            i++;
        }else{
            continue;
        }
    }
    return lotto;
}
```