

Java를 알고 C배우기

컴퓨터프로그래밍3

week 10-2 다차원 배열과 포인터- 다차원배열과 함수 인자

2022.1학기
충남대 조은선



배열포인터 vs. 포인터 배열 (복습)

```
int num1=10, num2=20, num3=30, num4=40;  
int arr2d[2][4]={1, 2, 3, 4, 5, 6, 7, 8};  
int i, j;
```

```
int * whoA [4];    // 포인터 배열  
int (*whoB) [4];  // 배열 포인터
```

```
int * whoA[4]={&num1, &num2, &num3, &num4};    // 포인터 배열  
int (*whoB)[4]=arr2d;    // 배열 포인터
```

```
printf("%d %d %d %d \n", *whoA[0], *whoA[1], *whoA[2], *whoA[3]);  
for(i=0; i<2; i++)  
{  
    for(j=0; j<4; j++)  
        printf("%d ", whoB[i][j]);  
    printf("\n");  
}
```

실행결과

```
10 20 30 40  
1 2 3 4  
5 6 7 8
```

whoB[i][j] ?

~~(*whoB)[i][j]?~~



2차원 배열을 함수의 인자로 전달

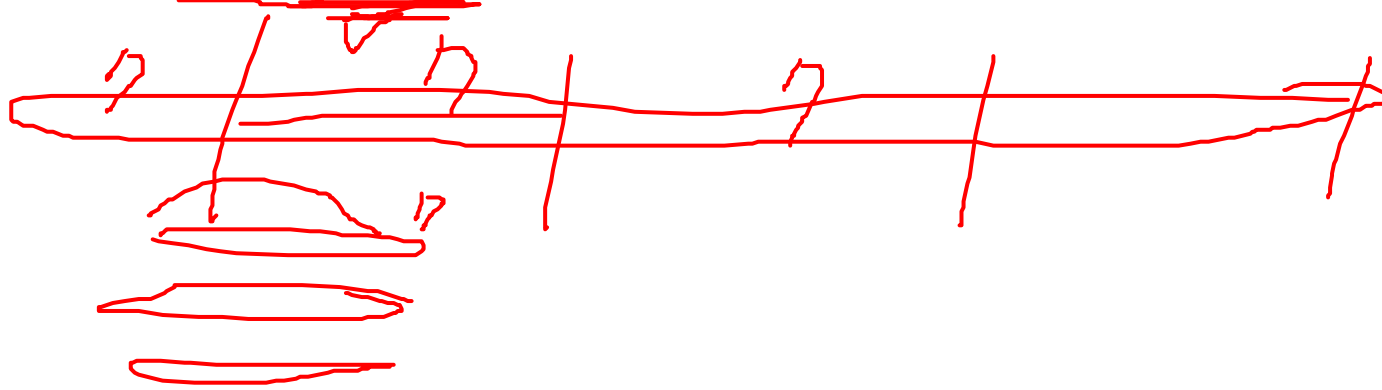
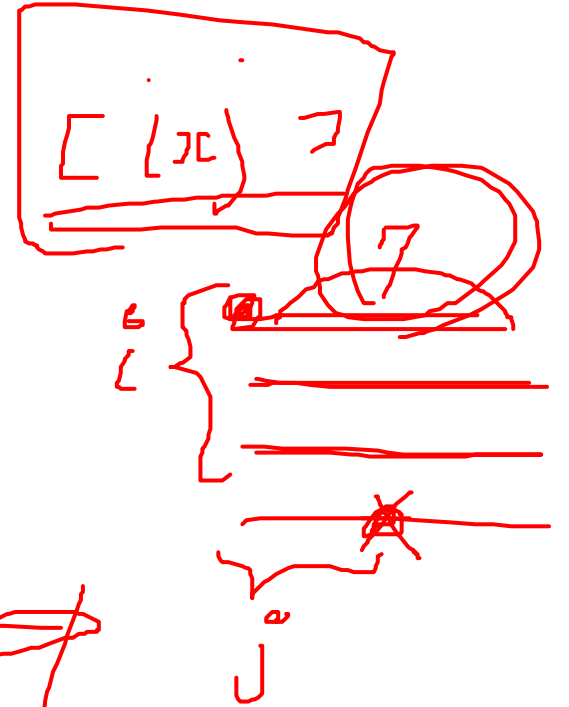
```
int arr1[2][7];           // int (*arr1)[7]
double arr2[4][5];        // double (*arr2)[5]
simple_func(arr1, arr2);
```

함수 선언

```
void simple_func(int (*p1)[7], double (*p2)[5]) {...}
```

또는

```
void simple_func(int p1[][7], double p2[][5]) {...}
```



예

```
void ShowArr2DStyle(int (*arr)[4], int column)
{
    // 배열요소 전체출력
    int i, j;
    for(i=0; i<column; i++)
    {
        for(j=0; j<4; j++)
            printf("%d ", arr[i][j]);
        printf("\n");
    }
    printf("\n");
}
```

```
int Sum2DArr(int arr[][4], int column)
{
    // 배열요소의 합 반환
    int i, j, sum=0;
    for(i=0; i<column; i++)
        for(j=0; j<4; j++)
            sum += arr[i][j];
    return sum;
}
```

```
1 2 3 4
5 6 7 8
```

```
1 1 1 1
3 3 3 3
5 5 5 5
```

```
arr1의 합: 36
arr2의 합: 36
```

실행결과

정의된 두 함수의 인자로 전달되는 2차원 배열의 **가로길이는 결정되어 있다.**

반면, **세로 길이 정보는 결정되어 있지 않고** 두 번째 인자를 통해서 추가로 전달하고 있다. 이점에 주목하자!

```
int main(void)
{
```

```
    int arr1[2][4]={1, 2, 3, 4, 5, 6, 7, 8};
    int arr2[3][4]={1, 1, 1, 1, 3, 3, 3, 3, 5, 5, 5, 5};
```

```
    ShowArr2DStyle(arr1, sizeof(arr1)/sizeof(arr1[0]));
```

```
    ShowArr2DStyle(arr2, sizeof(arr2)/sizeof(arr2[0]));
```

```
    printf("arr1의 합: %d \n", Sum2DArr(arr1, sizeof(arr1)/sizeof(arr1[0])));
```

```
    printf("arr2의 합: %d \n", Sum2DArr(arr2, sizeof(arr2)/sizeof(arr2[0])));
```

```
    return 0;
```

배열의 세로길이 계산방식



2차원 배열에서도 $arr[i] == *(arr+i)$

```
int arr[3][2] = {{1,2}, {3,4}, {5,6}};
```

arr[0]	1024		
arr[1]	1032		
arr[2]	1040		///

*arr	1024		
*(arr+1)	1032		
*(arr+2)	1040		///

arr에는 1024가 들어감

arr[1] == *(arr+1)

arr + 1 == 1032, arr[1] == 1032

// arr[1]은

// 한 행을 나타내는 포인터들

arr[2][1] = 7; 와 동일한 표현 세가지

*(arr+2)[1] = 7;

// *(arr+2) : 두번째 행 시작주소, (== arr[2])

// [1] : 거기서 1번째 요소의 내용에 7을 지정

*(arr[2] + 1) = 7;

// arr[2] : 두번째 행 시작주소

// +1 : 거기서 1 증가, 즉 그 행의 1번째 요소의 주소

// *: 그 내용에 7을 지정

((arr+2) + 1) = 7;

// *(arr+2) : 두번째 행 시작주소 (== arr[2])

// +1 : 거기서 1 증가, 즉, 그 행의 1번째 요소의 주소

// * : 그 내용에 7을 지정



Quiz

```
int Sum2DArr(_____, int n) {  
    for (int i = 0; i < column; i++) {  
        for (int j = 0; j < 4; j++)  
            printf("%d", arr[i][j]);  
        printf("\n");  
    }  
    printf("\n");  
}
```

위와 같은 함수의 밑줄에 들어갈 수 있는 것으로 알맞은 것을 모두 고르시오.

- (1) int arr[][]
- (2) int (*arr) []
- (3) int (* arr)[3]
- (4) int arr[][3]
- (5) int arr[3][]

