

11주차 3차시 다차원 배열 및 함수 포인터 실습

【학습목표】

1. 실습을 통하여 다차원 배열을 학습하고 실행할 수 있다.
2. 실습을 통하여 함수 포인터를 학습하고 실행할 수 있다.

학습내용1 : 다차원 배열 실습

- 2차원 배열의 위치에 대한 예제 실습 (교재 11주차 1차시 예제 참조)
도서 source: 2DArrayAddress.c
- 2차원 배열의 포인터 형을 결정짓는 예제 실습 (교재 11주차 1차시 예제 참조)
도서 source: 2DArrayNameAndArrPtr.c
- 2차원 배열을 함수의 인자로 전달하기 예제 실습 (교재 11주차 1차시 예제 참조)
도서 source: 2DArrayParam.c

- 아래 코드에서 ???을 대신할 수 있는 매개변수 선언을 추가하시오

```
void ComplexFuncOne(???,???) { . . . }  
void ComplexFuncTwo(???,???) { . . . }  
int main(void)  
{  
    int* arr1[3];  
    int* arr2[3][5];  
    int** arr3[5];  
    int*** arr4[3][5];
```

```
    ComplexFuncOne(arr1, arr2);  
    ComplexFuncTwo(arr3, arr4);
```

학습내용2 : 함수 포인터 실습

- 적절한 함수 포인터 변수의 선언 예제 실습(교재 11주차 2차시 예제 참조)
도서 source: FunctionPointer.c
- 형(Type)이 존재하지 않는 void 포인터 예제 실습(교재 11주차 2차시 예제 참조)
도서 source: VoidTypePointer.c

【학습정리】

1. 이차원 배열 `int arr2d[3][3];`에서 배열이름 `arr2d`가 가리키는 것은 인덱스 기준으로 `[0][0]`에 위치한 첫 번째 요소이다.
2. 함수의 이름은 함수가 저장된 메모리 공간을 가리키는 포인터이다(함수 포인터).
3. 함수의 이름이 의미하는 주소 값은 함수 포인터 변수를 선언해서 저장할 수 있다.
4. 데이터의 입출력이 가능한 이유는 출력의 경로가 되는 출력 스트림과 입력의 경로가 되는 입력 스트림이 존재하기 때문이다.