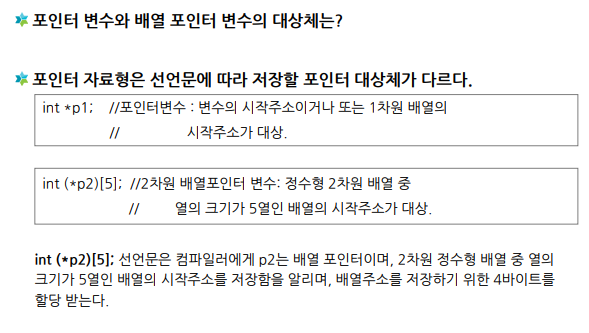
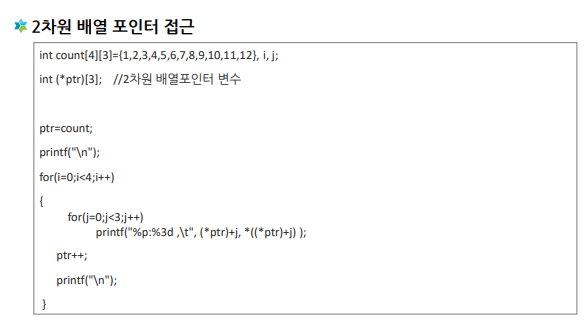


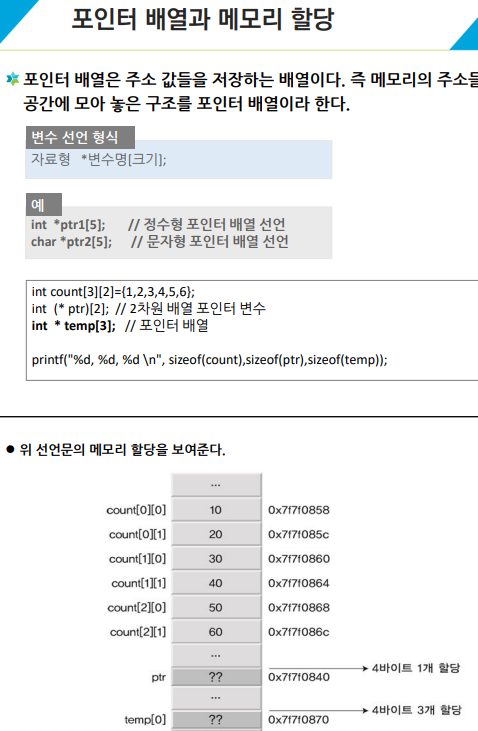
Int \*a1[3]; // 포인터 배열

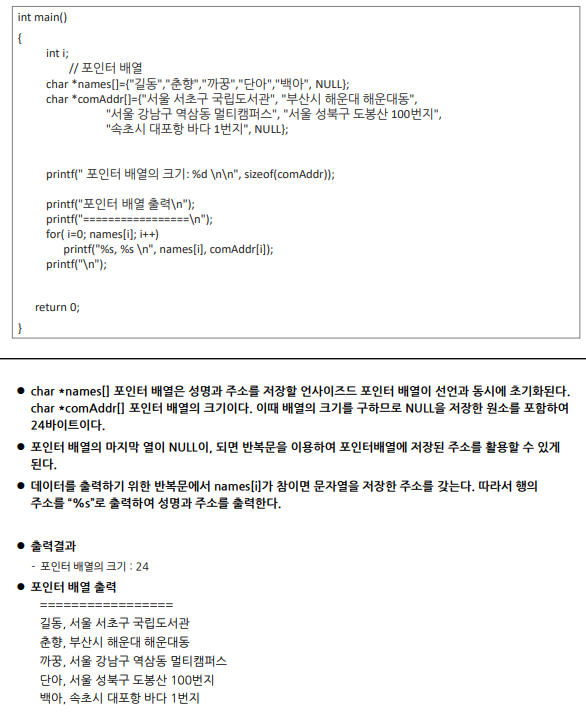
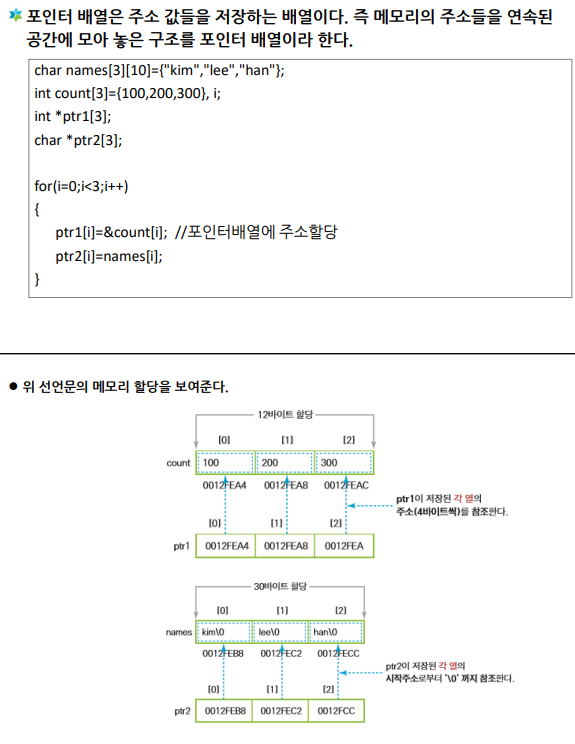
Int (\*a2)[3]; // (2차원)배열포인터, 4

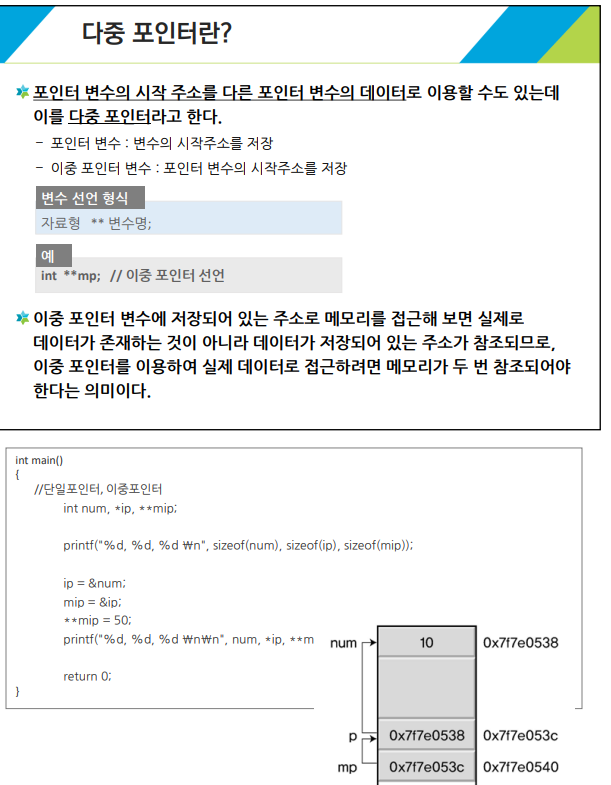
Int (\*a3)(int); // 함수포인터 변수, 4









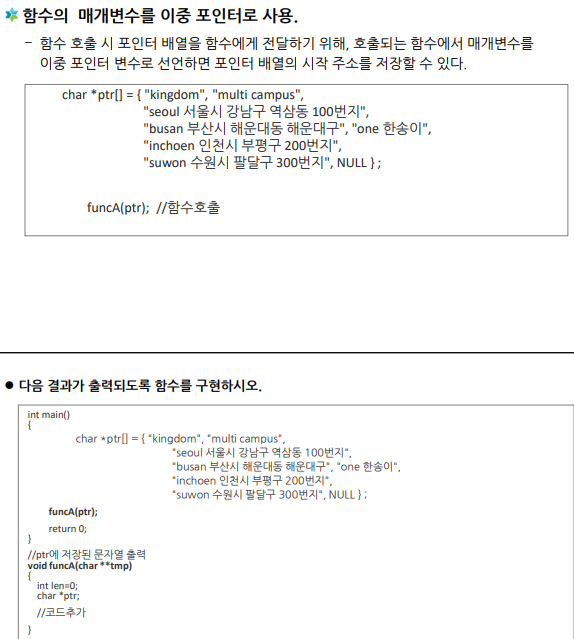


Num ip mip

|500| |3000| |4000| // value

*3000 4000 5000 //address*

# 이중 포인터 필요성: heap 세그먼트 사용할 때 필요



Ptr 0 1 2 3 4 5 6 7

|7000| |7100| |7200| |7300| |7400| |7500| |7600| |7700|

*4000 4008 4016 4032 4040 4048 4056 4064*

kingdom multi campus hello world one 하나 two 둘 seoul 서울 busan 부산 NULL

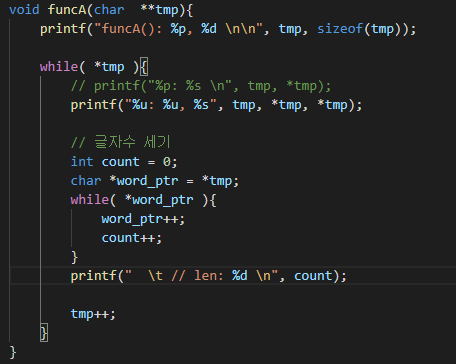
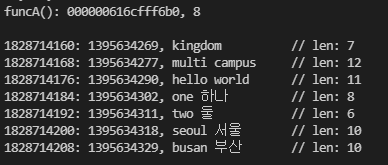
*7000 7100 7200 7300 7400 7500 7600 7700*

Tmp |4000|

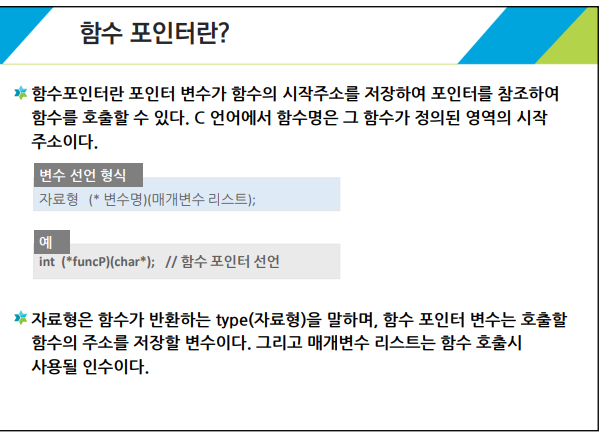
*5000*

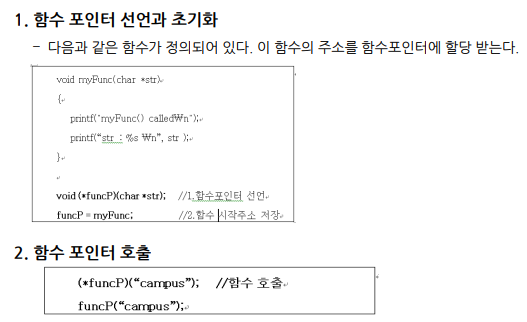
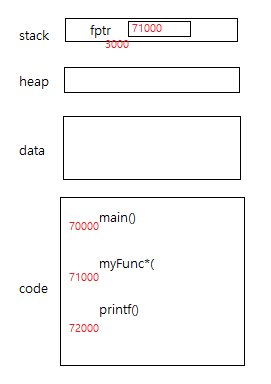
Printf(“%c \n”, \*\*) // 그 문자열의 첫번째 글자가 나옴(배열로 취급)

# 추가미션 : 문자열배열에서 나오는 문자열 길이 세기



# 함수포인터 필요성: 콜백함수에서 반드시 필요





Int (\*a2)[3]; // (2차원)배열포인터, 4

Int (\*a3)(int); // 함수포인터 변수, 4

**Func\_pointer2.c**

* **함수포인터 자료형(typedef) 활용하기**
* **함수포인터 자료형으로 함수포인터 배열 활용하기**

typedef int (\*FP)(int, int);

함수포인터를 typedef로 선언하면 함수포인터 변수가 아닌, 함수포인터 자료형이 된다.

