1. test_pointer_size.c

```
#include <stdio.h>

int main()
{
	{
		int ar1[20][10];
		printf("ar1: %lu %lu %lu₩n", sizeof(ar1), sizeof(*ar1), sizeof(**ar1));
	}
	{
		int (*k)[10]; //  int [10] 의 주소
		printf("k: %lu %lu %lu₩n", sizeof(k), sizeof(*k), sizeof(**k));
	}
	{
		int *j[10];  // int * 의 배열
		printf("j: %lu %lu %lu₩n", sizeof(j), sizeof(*j), sizeof(**j));
	}
	{
		int **j;
		printf("j: %lu %lu %lu₩n", sizeof(j), sizeof(*j), sizeof(**j));
	}
}
```

2. test_pointer_compatibility.c

```
#include <stdio.h>

int main()
{
	// 주석 없는 줄은 문제 없는 코드.
	int *pt;
	int (*pa)[3];
	int ar1[2][3];
	int ar2[3][2];
	int **p2;
	pt=&ar1[0][0];
	pt=ar1[0];
	//pt=ar1;    // ar1  은 이중 배열이라, int 3 개짜리 배열의 주소(int (*)[3])와 호환됨.
	pa=ar1;
	//pa=ar2; // ar2 는 int(*)[2]와 호환됨.
// 좀더 자세히 설명하면, 주소 자체는 integer 의 주소로  볼수도 있지만,  ar+1 은 ar 보다 8byte 뒤. pa+1 은 pa 보다 12byte 뒤라서 호환이 안됨.  두 변수 모두 int*로 강제 형변환은 문제없이 가능함.    즉, pt=(int*)ar2; 는 문제 없음.
	p2=&pt;  // 이 줄이 없으면, 아래 두줄이 컴파일 시점에는 문제가 없지만 실행시 문제가 될 수 있음.
	*p2=ar2[0];
	*p2=* ar2;
	//p2=ar2;  // ar2 는 int (*)[2] 라서 강제로 형변환을 해도 int *이지 int ** 이 아님. *ar2 는 int[2] 로 int* 와 호환된다는 점때문에 혼동하기 쉬움.
```

```
        return 0;
}
```