

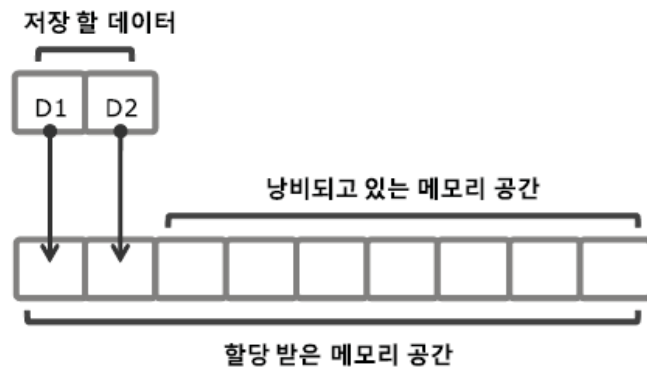
전화번호 관리 프로젝트 **3**단계

변경하자! 구조체 포인터 배열로

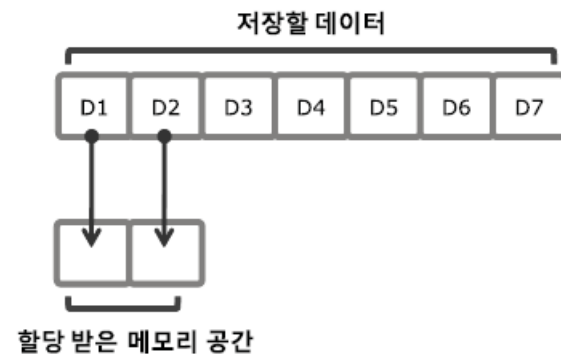


www.orentec.co.kr

구조체 배열의 장점과 단점

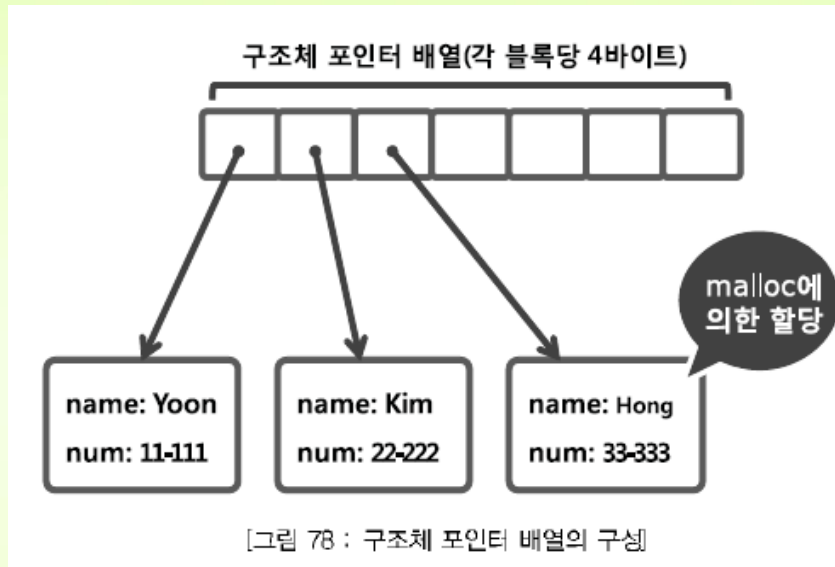


[그림 76 : 낭비되고 있는 메모리 공간]



[그림 77 : 부족한 메모리 공간]

구조체 포인터 배열의 장점과 단점



문제의 제시

업데이트 및 변경 포인트 1 : 구조체 배열을 구조체 포인터 배열로 변경

소스파일 phoneFunc.c에 전역으로 다음과 같이 선언된 구조체 배열이 존재한다.

```
phoneData phoneList[ LIST_NUM ] ;
```

이 구조체 배열을 다음과 같이 구조체 포인터 배열로 변경하고, 이에 따라서 더불어 변경 및 추가되어야 할 모든 요소들을 적절히 변경 및 추가하여라.

```
phoneData * phoneList[ LIST_NUM ] ;
```

업데이트 및 변경 포인트 2 : 출력 관련 함수 추가

screenOut.c에 다음과 같이 구조체 변수를 인자로 전달받아서 적절한 출력을 구성하는 함수가 정의되어 있다.

```
void ShowPhoneInfo(phoneData phone){ ...}
```

이 함수를 계속해서 사용하는 것도 문제되지 않으나, 배열이 구조체 포인터 배열인 만큼 구조체 포인터를 인자로 받는, 다음과 같은 함수를 정의할 필요가 있다.

```
void ShowPhoneInfo(phoneData * pPhone){ ...}
```

그러나 이 함수를 추가로 정의했다 하여 기존에 정의한 함수를 삭제할 필요는 없다. 함수 자체에 문제가 없다면 그냥 남겨두는 것이 향후의 변경에 빠르게 대응할 수 있는 길이기도 하다. 그래서 기존에 정의한 함수를 남겨두기 위해 함수의 이름을 다음과 같이 정의하겠다.

```
void ShowPhoneInfoByPtr(phoneData * pPhone){ ...}
```

끝으로 정의를 했으면 사용해야 할 것 아닌가? 이전에 ShowPhoneInfo 함수를 호출하던 부분을 ShowPhoneInfoByPtr 함수를 호출하는 형태로 변경하자.

