



# **Programming - Practice 08**

- **file I/O**

# ▶ Phone book file I/O

## ■ 1. 주소록을 파일로 저장하여 관리한다.

- (1) 주소록 정보를 저장하는 구조체 `Sperson`의 배열을 파일에 저장하고 정보를 파일에서 읽어온다.
- (2) 주소록을 이름 순으로 정렬한다.
- (3) 다음과 같이 메뉴를 추가

1. Add a person
  2. Print the list
  3. Name search
  4. Birthday this month!
  5. Sort the phone book
  6. Save the phone book
  7. Load the phone book
  0. Quit the program
- Enter the command:

# ▶ Phone book file I/O

- 파일 쓰기 및 읽기
  - text file 또는 binary file
  - 읽는 형식과 쓰는 형식이 일치해야 함
- Text file
  - 키보드 입력과 유사한 형식으로 저장
    - 각 항목을 한 줄에 텍스트로 저장
  - 생년월일 자리 수 형식
    - ex) 1990. 2. 4 -> 19900204
    - fprintf() 형식에서 "%02d"
  - 읽기는 문자열 세 개 읽어 처리
    - 파일 마지막인지 각각 확인

```
[1] Name: Steve Jobs  
    Telephone: 010-1234-5678  
    Birthday: 1955. 2. 24.
```

위 입력을 텍스트 파일에 저장하면  
다음과 같이 저장된다.

```
Steve Jobs  
01012345678  
19550224
```

# ▶ Phone book file I/O

- Binary file

- fread()와 fwrite() 사용하여 저장

- 장 단 점

- 텍스트는 편집기로 수정 및 확인 가능하지만 잘 못 수정할 경우 읽기 형식에 오류가 생길 수 있다. 또한 저장 크기가 일정하지 않아 임의의 위치 접근이 어려움
  - 이진파일(binary file)은 각 항목 당 저장 크기가 일정하여 임의적 위치 접근이 쉽다

- 어떤 형식으로 할지는 선택

- 파일 이름은 상수로 설정

# ▶ Phone book sorting

- 이름 오름차순으로 리스트 정렬을 한다
- 라이브러리의 `qsort()` 함수 이용(`stdlib.h`)
  - 가장 빠른 알고리즘 중의 하나인 quick sort 구현

```
void qsort (void* base, size_t num, size_t size,  
            int (*compar)(const void*,const void*));
```

- 아이템 크기 비교를 위한 함수를 작성해야 함

```
int values[] = { 40, 10, 100, 90, 20, 25 };  
int compare (const void *a, const void *b)  
{  
    return (*(int *)a - *(int *)b);  
}  
qsort (values, 6, sizeof(int), compare);
```

  - 문자열은 `strcmp` 함수 이용하여 비교 함수 작성

리턴 값	의미
<0	*a < *b
0	*a == *b
>0	*a > *b