



# **Programming - Practice 06**

- **phone book**

# ▶ 1. Phone book

## ■ 1. 주소록 관리 프로그램을 작성한다.

- (1) 주소록 정보를 저장하는 구조체 정의

```
typedef struct struct_date {  
    int year, month, day;  
} Sdate;
```

```
typedef struct struct_person {  
    char name[MAX_STR];  
    char telephone[MAX_STR]; // 전화번호(문자열로 저장)  
    Sdate birthday;  
} Sperson;
```

- (3) Sperson의 배열을 만들어 입력을 받아 출력한다.

## ▶ 정보 입력

- 이름: 문자열 입력
- 전화번호: 핸드폰번호로 가정하며 '-'제외하고 숫자만 문자열로 입력을 받는다. 유효한 입력인지 확인한다.
- 생년월일: 문자열로 **yyyymmdd** 형태로 입력받아 숫자로 전환하여 저장한다. 유효한 입력인지 확인.
  - 예) "19550224"를 입력받아 숫자 1955, 2, 24로 바꾸어 birthday의 year, month, day에 저장한다.
- 입력 예:  
Enter the name: Steve Jobs  
Enter the phone number: 01012345678  
Enter the birthday (yyyymmdd): 19550224  
  
Name: Steve Jobs  
Telephone: 01012345678  
Birthday: 1955. 2. 24.

# ▶ 정보 입력

## ■ 유효한 입력

- 이름: 모두 허용
- 전화번호: 모두 번호인 경우만 유효
- 생년월일:  $\text{year} \geq 0, 1 \leq \text{month} \leq 12, 1 \leq \text{day} \leq 31$
- 잘못된 입력이 있을 수 있지만 간단한 경우만 확인해 보자

## ■ 모듈화된 코드 권장

- 작업을 쪼개어 적절한 함수를 만들어 구현하는 것을 권장
- 함수로 기능을 구현할 경우 나중에 재사용도 용이함
- 다음 실습 시간에 재사용할 것이니 코드 보관 바랍니다

# ▶ Challenge problem

- 전화 번호 출력: 휴대폰 및 일반전화의 숫자만 문자열로 입력된 것을 '-'을 추가하여 출력한다.
- 규칙
  - 첫 문자가 '0'일 경우 "01X", "02", "0XX" 다음에 '-'를 출력
  - 그 다음은 7자리일 경우 "XXX-XXXX"로 출력, 8자리 이상일 경우 "XXXX-XXXX"로 출력
- 예) 저장된 내용 → 출력 결과
  - 028200001 → 02-820-0001
  - 01012345678 → 010-1234-5678
  - 0111234567 → 011-123-4567
  - 0329876543 → 032-987-6543