



Programming - Term Project

- **calendar & schedule**

▶ Calendar & Schedule

■ Calendar & Schedule 프로그램

- 화면 구성: 달력 출력, 현재 달의 일정 출력, 명령어 표시

```
2022 November
Sun Mon Tue Wed Thu Fri Sat
    1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

Schedule
2022.11.21 01:00 PM 프로그래밍 수업
2022.11.29 03:00 PM 프로젝트 미팅

[ A: add a schedule, D: delete a schedule ]
[ LEFT: month--, RIGHT: month++, DOWN: year--, UP: year++, ESC: exit]
```

- 명령어

```
[ A: add a schedule, D: delete a schedule ]
[ LEFT: month--, RIGHT: month++, DOWN: year--, UP: year++, ESC: exit]
```



▶ Calendar

■ 달력을 출력한다

- 프로그램을 시작하면 오늘 날짜를 알아내어 이번 달 달력을 출력한다

```
#include<time.h>
```

```
time_t curr_t;  
struct tm *curr_time;  
int year, month;
```

```
time(&curr_t);  
curr_time = localtime(&curr_t);  
year = curr_time->tm_year + 1900;  
month = curr_time->tm_mon + 1;
```

- 화살표를 이용하여 다른 달의 달력을 출력한다

- 왼쪽: 저번 달, 오른쪽: 다음달, 위: 1년 후, 아래: 1년 전

- ESC: 종료

- getch()를 통해 키 입력을 알아낸 후 그 값에 따라 실행

- LEFT: 75, RIGHT: 77, UP: 72, DOWN: 80, ESC: 27

- 현재 달이 몇 일까지 있는지, 1일이 무슨 요일인지 알아야 함



달에 몇 일까지 있는가?

- `int year, int month, int day`가 연월일을 나타낸다고 하자.
날짜의 유효한 값은 다음 조건에 맞아야 한다.
 1. `year`는 1800 이상이다.
 2. `month`는 1~12의 값이다.
 3. `day`는 1, 3, 5, 7, 8, 10, 12월의 경우 1~31이고, 4, 6, 9, 11월의 경우 1~30이고, 2월은 평년은 1~28, 윤년은 1~29이다.
 4. 4로 나누어 떨어지는 해는 우선 윤년으로 한다. (2004, 2008, 2012, ...)
 5. 100으로 나누어 떨어지는 해는 평년으로 한다. (2100, 2200, 2300, ...)
 6. 400으로 나누어 떨어지는 해는 다시 윤년으로 한다. (1600, 2000, 2400, ...)

특정 날의 요일은 무엇인가?

- 특정 날짜의 요일을 계산하는 방법으로 다음과 같은 Zeller's congruence라는 방법이 있다. 이 때 year는 년도, month는 월, day는 일이다.

$$h = \left(q + \left\lfloor \frac{26(m+1)}{10} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor + 6 \left\lfloor \frac{Y}{100} \right\rfloor + \left\lfloor \frac{Y}{400} \right\rfloor + 6 \right) \% 7$$

- h : 요일 (0 = 일요일, 1 = 월요일, ..., 6 = 토요일)
- q : 일
- m : 월 (3 = 3월, 4 = 4월, 5 = 5월, ..., 12 = 12월, 13 = 1월, 14 = 2월)
- Y : 3~12월은 년도 그대로, 1~2월은 년도-1

예) 2022년 2월 => Y = 2021, m = 14

2022년 10월 => Y = 2022, m = 10



▶ Schedule

- 스케줄 저장을 위한 구조체 설계
 - 스케줄 시작 시간 정보를 저장한다: 연도, 월, 일, 시, 분
 - 시는 24시간 표시 기준으로 저장하고(0~23) 출력할 때는 12시간 표시 기준으로 출력한다(13 -> 1 PM)
 - 스케줄 내용을 문자열로 저장하는 멤버 변수가 필요하다
- 스케줄 배열
 - 스케줄을 배열에 저장한다 (정적 배열 또는 동적 배열 선택 가능)
 - 스케줄은 배열에 시간 순으로 저장한다
 - 스케줄을 새로 추가할 경우 해당 위치를 찾아 추가해준다
- 스케줄 출력
 - 달력을 출력할 때 해당 월의 스케줄을 모두 출력한다



▶ Schedule

- schedule을 정렬된 상태로 관리
 - 날짜와 시간이 먼저인 것이 앞에 오도록 관리
 - 스케줄을 추가할 때 알맞은 자리를 찾아 넣는다 (배열을 사용하므로 해당 자리 이후의 것은 뒤로 한 칸씩 이동해야 함)
 - 해당 자리를 찾기 위해 배열의 처음부터 시작하여 추가할 스케줄의 날짜보다 작은 것은 모두 건너 뛰어 추가할 자리를 찾는다
 - 비교할 때는 년, 월, 일, 시, 분 순서대로 비교한다
 - * 일단 맨 뒤에 추가하고 정렬해도 된다
- 스케줄을 프로그램 끝날 때 파일에 저장하고 프로그램 시작할 때 파일에서 읽는다

▶ Add Schedule

- 스케줄을 배열에 추가한다
 - 현재 달력에서 출력되고 있는 year와 month에 추가
 - day, hour, min, note는 사용자 입력
 - day, hour, min의 입력 값이 유효한 값인지 확인한다
 - 함수를 통하여 해당 달에 몇 일까지 있는지 확인
 - hour는 0~23, min은 0~59
 - 실행 예

```
### Add a schedule ###  
Day: 14  
Hour(0~23): 13  
Min(0~59): 0  
Note:  
Programming
```



2022 December

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

schedule

```
2022.12.14 01:00 PM Programming  
2022.12.19 01:00 PM Final exam
```


▶ Delete Schedule

- 스케줄을 배열에서 삭제한다
 - 삭제된 위치 뒤의 요소들을 한 칸씩 앞으로 이동해야 한다
 - 해당 달의 일정을 일련 번호와 함께 출력하여 보여주고 그 중에서 삭제할 번호를 선택한다
 - 입력이 범위를 벗어날 경우 삭제를 취소한다

Schedule

1. 2022.12.14 01:00 PM Programming
2. 2022.12.19 01:00 PM Final exam
3. Cancel



2022 December

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

which schedule do you want to delete? 1
Schedule deleted.

Schedule

2022.12.19 01:00 PM Final exam

▶ Calendar & Schedule

■ 여러 개의 파일로 구성해보자

□ 예시

