projects\project3\solution\main.py

```
# SOLUTION
##########################################################################################
# 프로젝트 설명
# - 길동이는 2023년 6월부터 만나이를 사용한다는 것을 알고 혼란스러웠다.
#   한국식 나이에 익숙해져 만나이를 계산하는 것이 번거롭고 같은 해에
#   태어난 친구끼리 생일이 지났느니 안지났느니 따지는 것도 복잡하게 느껴졌다.
#
# - 그래서 그냥 살아온 날 수로 나이를 셈하는 편이 제일 정확하다고 생각했기 때문에
#   길동이는 파이썬으로 태어난 년, 월, 일을 입력받고 현재 또는 특정 날의 년, 월, 일을 입력받아
#   두 시점의 날 수 차이를 계산하는 프로그램을 만들고 싶었다.
#
# - 다음 조건을 만족 시키면서 해당 프로그램을 작성하시오.
#     - 입력은 반드시 숫자로만 주어져야 한다.
#       반복문을 적절히 사용하여 사용자가 올바른 입력을 할때까지 입력을 받도록 작성
#     - 문제를 간단히 하기 위해 뒤에 입력하는 연도는 앞에 입력하는 연도보다 적어도 1 커야 한다.
#       즉 동일한 연도에 두 날짜 차이는 계산하지 않는다.
#       (실제로 1년도 살지 않은 아기의 나이를 비교할 일은 없으니까!)
#     - 날 수를 계산할 때 반드시 윤년을 검사해서 적절히 1일 씩 더 해주어야 함.
#       이를 위해 이미 구현된 `is_leap_year()`와 `get_month_days()`가 제공되니
#       주석을 읽고 이 함수들을 활용할 것
##########################################################################################

from datetime import date

# 프로젝트를 완성하기 위해 별도 라이브러리 함수를 사용
# is_leap_year(year) 형식으로 호출하면 year가 윤년이면 True, 아니면 False를 반환
# get_month_days(year, month) 형식으로 호출하면 해당 month의 날 수를 반환
# Mission 1에서 완성되어야 함.
from days_utils import MONTH_DAYS, is_leap_year, get_month_days

#########################################################################
# Mission 2 : 두 시점의 차이를 계산하는 함수를 작성하시오.
#             태어난 해에 산 날 수를 구해서 days_part_1에 저장
def my_days(birth_year, birth_month, birth_day, cur_year, cur_month, cur_day):
    # birth_year: 태어난 해
    # birth_month: 태어난 달
    # birth_day: 태어난 날
    # cur_year: 현재 해 또는 차이를 계산하고 싶은 특정 해
    # cur_month: 현재 달 또는 차이를 계산하고 싶은 특정 달
    # cur_day: 현재 날 또는 차이를 계산하고 싶은 특정 날

    # birth_year에 산 날 수를 계산하여 이 변수에 저장
    days_part_1 = 0

    # step 1: 태어난 해의 태어난 달에 산 날 수를 계산해서 days_part_1에 저장
    #         태어난 당일은 포함하지 않는다.
    days_part_1 += MONTH_DAYS[birth_month] - birth_day # 태어난날 포함이면 +1

    # birth_year가 윤년인지를 판단하고 윤년이면 +1
    # is_leap_year()를 사용하여 윤년인지 판단하시오.
    if birth_month == 2 and is_leap_year(birth_year):
```

```
        days_part_1 += 1

    # step 2: 태어난 해의 태어난 달 다음달 부터 12월 31일까지 날을 계산해서
    #         days_part_1에 더함
    #         range에 적당한 범위를 설정하시오.
    #         step 2에서는 step1과 다르게 Mission 1에서 완성한 get_month_days()를 사용하여
    #         해당 년도, 월의 날을 윤년을 고려하여 안전하게 가져오시오.
    for m in range(birth_month+1, 13):
        days_part_1 += get_month_days(birth_year, m)
    ###########################################################################


    ###########################################################################
    # Mission 3: 태어난 해 다음 해 부터 현재 해 이전 해까지 산 날 수를 계산해서
    #            days_part_2에 저장
    #            [!주의!] 반드시 윤년을 체크하여 반영하시오.
    days_part_2 = 0

    # 바깥 for 루프에서 년도를 y로 순회하고
    for y in range(birth_year+1, cur_year):
        # 안쪽 for 루프에서 y 년도의 달을 m으로 순회하면서
        for m in range(1,13):
            # days_part_2에 m월의 날 수를 합산
            days_part_2 += get_month_days(y, m)

    ###########################################################################
    # 현재 해 산 날 수를 계산해서 days_part_3에 저장
    #
    days_part_3 = 0

    # step 1: 태어난 해와 현재 해가 다른 경우만 다루기 때문에
    #         1월 부터 현재 달 이전 달까지 산 날수를 모두 계산해서
    #         days_part_3에 저장
    for m in range(1, cur_month):
        days_part_3 += get_month_days(cur_year, m)

    # step2: 현재 달의 산 날 수를 days_part_3에 더함
    days_part_3 += cur_day

    # 세 개 파트로 나눠서 계산한 날 수를 합산해서 반환하시오.
    return days_part_1 + days_part_2 + days_part_3
    ###########################################################################


#####################################################################
# 검증 함수
# python의 datetime.date를 사용해서 정확한 날 수를 계산
def true_days(birth_year, birth_month, birth_day, cur_year, cur_month, cur_day):
    d1 = date(birth_year, birth_month, birth_day)
    d2 = date(cur_year, cur_month, cur_day)
    delta = d2 - d1

    return delta.days # 태어난날 포함이면 +1
```

```
# 여기서부터 시작해서 주석처리 (Mission 4를 수행하기 전에)
if __name__ == '__main__':
    ##########################################################################################
    # Mission 5: 적절한 형태로 사용자 입력 받기
    # [!주의!]: Mission 4를 테스트 하기 전에 Mission 5 부분을 주석 처리하시오.
    #
    # 다음 입력을 지정된 변수에 입력받으시오.
    # 태어난 연도: birth_year
    # 태어난 달: birth_month
    # 태어난 날: birth_day
    # 현재 연도: cur_year
    # 현재 달: cur_month
    # 현재 날: cur_day
    #
    # 조건1: 모든 입력은 반드시 숫자만 입력받아야 하며 숫자가 아닌 문자가 입력되면
    #        "숫자만 입력하세요."를 출력하고
    #        숫자가 입력될 때까지 반복해서 입력받음
    #
    # 조건2: birth_year(태어난 연도)보다 cur_year(현재 연도)가 더 커야 함
    #        그렇지 않으면
    #        "현재 연도는 태어난 연도 보다 커야 합니다."를 출력하고
    #        반복해서 현재 연도를 입력받음
    #
    # step 1, 2, 3을 완성하고 프로그램을 실행하여 의도한대로 입력되는지 확인하시오.
    #
    # step 1
    # 모든 입력을 숫자로 입력받기 위해 즉 조건1을 만족시키기 위해
    # filter_int_input 함수를 작성하시오.
    # hint: user_input 변수가 숫자인지 구별하는 함수 user_input.isidigit()
    def filter_int_input(msg):
        # 숫자가 온전히 입력될 때까지 루프를 돈다.
        while True:
            user_input = input(msg)

            # 적당한 조건문을 사용하시오.
            if not user_input.isdigit():
                print("숫자만 입력하세요.")
            else:
                user_input = int(user_input)
                break

        return user_input

    # step2 각 변수에 사용자 입력을 받아 할당하시오.
    #       input() 함수 대신 위에서 작성한 filter_int_inpu()을 사용하시오.
    birth_year = filter_int_input("태어난 연도를 입력: ")
    birth_month = filter_int_input("태어난 달을 입력: ")
    birth_day = filter_int_input("태어난 일을 입력: ")

    # step3: 조건2 에 맞을 때 까지 현재 연도, 월, 일를 입력받으시오.
    # hint: continue 문을 적절한 위치에 삽입하시오.
    while True:
        cur_year = filter_int_input("현재 연도를 입력: ")
        if cur_year <= birth_year:
```

```
            print("현재 연도는 태어난 연도 보다 커야 합니다.")
            continue

        cur_month = filter_int_input("현재 달을 입력: ")
        cur_day = filter_int_input("현재 일을 입력: ")

        break
# 여기까지 주석처리 (Mission 4를 수행하기 전에)

    ##########################################################################################
    # Mission 4: 검증, 아래 두 출력 결과는 동일해야 함
    # 위 Mission 5 부분을 전체 주석처리하고
    # my_days()에 적당한 연도, 월, 일을 직접 입력하고
    # 동일한 숫자를 true_days()에 입력해서 두 결과가 동일하기 나오는지 검증하시오.
    #
    print('내가 계산한 날 수      : ', my_days(birth_year, birth_month, birth_day, cur_year,
cur_month, cur_day) )
    print('datetime이 계산한 날 수: ',  true_days(birth_year, birth_month, birth_day, cur_year,
cur_month, cur_day) )

    # 동일한 숫자가 출력되면 이 파일과 같은 폴더에 있는 test.py를 실행해서
    # 10만개 무작위 날짜에 대해서 테스트를 수행하시오.
    # Mission 4가 성공적으로 통과되었으면 Mission 5를 수행하시오.
    # Mission 5를 수행하기 전에
    # 직접 입력한 연도, 월, 일을 다시 birth_year, birth_month, birth_day
    # cur_year, cur_month, cur_day로 돌려놓으시오.
```