Abstract Data Structures

During the lecture you saw the **Date** class in the context of date abstraction data type. The goal of this assignment is to expand that Abstract Data Structure by adding functionality. Remember to reason about abstraction while implementing it, use what is provided.

Assignment 0

Transcribe the example as two Python file:

- **main.py** - containing a main body that will use the implemented functions
- **date.py** - containing the example provided

Is important that you read the code that has been provided and you understand its content.

ATTENTION : there may be a few mistakes in the code of the book, fix them!

Assignment 1

Complete the partial implementation of the Date class by implementing the remaining methods:

1. **monthName**() - Returns the Gregorian month name of this date.
2. **isLeapYear**() - Determines if this date falls in a leap year and returns the
3. appropriate boolean value.
4. **numDays**() - Returns the number of days as a positive integer be-
5. tween this date and the otherDate.
6. **advanceBy**() - Advances the date by the given number of days. The date is incremented if days is positive and decremented if days is negative. The date is capped to November 24, 4714 BC, if necessary.
7. **isValidGregorian**() - return whether the three components of a date are valid (month, day, year)

ATTENTION : dates are written using the American convention, i.e. Month, Day, Year.

Assignment 2

Further expand the Date class by adding these methods:

1. **dayOfWeekName**() - returns a string containing the name of the day.
2. **dayOfYear**() -returns an integer indicating the day of the year. For example, the first day of February is day 32 of the year.
3. **isWeekday**() - determines if the date is a weekday.
4. **isEquinox**() - determines if the date is the spring or autumn equinox (use as dates March 20th and September 22nd)
5. **isSolstice**() - determines if the date is the summer or winter solstice (use as dates June 21st and December 21st)

6. **asGregorian**(divchar = '/') - similar to the str() method but uses the optional argument divchar as the dividing character between the three components of the Gregorian date.

Assignment 3

Implement a function named **printCalendar**() that accepts a Date object and prints a calendar for he month of the given date. For example, if the Date object passed to the function contained the date 1/31/2021, the function should print:

```
              January 2021
Su    Mo    Tu    We    Th    Fr    Sa
                              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
```

Assignment 4

Modify the Date() constructor to make each of the three arguments optional, with an initial value of *None* .

When no argument is supplied to the constructor, the object should be initialized to the current date.

Hint: You will need to use Python's `today()` function from the `datetime` module.

For example (supposing the current day is 1/1/2021:

```
>>> date = Date(year=2001)
>>> print(date.asGregorian())
1/1/2001
```