import event

class Day:
    """A calendar day and its events.""

def __init__(self, day, month, year):
    """ (Day, int, str, int) -> NoneType

    Initialize a day on the calendar with day, month and year, and no events.
    
    >>> d = Day(25, 'November', 2015)
    >>> d.day
    25
    >>> d.month
    'November'
    >>> d.year
    2015
    >>> d.events
    []
    ""

    def schedule_event(self, new_event):
        """ (Day, Event) -> NoneType

        Schedule new_event on this day, even if it overlaps with an existing event. Later we will improve this method.
    
    >>> d = Day(27, 'November', 2015)
    >>> e = event.Event(11, 12, 'Meeting')
    >>> d.schedule_event(e)
    >>> d.events[0] == e
    True
    """
def __str__(self):
    """ (Day) -> str

    Return a string representation of this day.
    """

>>> d = Day(8, 'December', 2015)
>>> d.schedule_event(event.Event(15, 16, 'Submit A3 work'))
>>> d.schedule_event(event.Event(16, 23, 'Celebrate end of classes'))
>>> print(d)
8 December 2015:
- Submit A3 work: from 15 to 16
- Celebrate end of classes: from 16 to 23
"""

if __name__ == '__main__':

    # Create day 16 December 2015.

    # Add an event "Sleep in" from 0 to 11 on 16 December 2015.

    # Add an event "Brunch" from 11 to 13 on 16 December 2015.

    # Print the day 16 December 2015, including its events.