For each method, implement the body.

```python
import event

class Day:
    """A calendar day and its events.""
    
    def __init__(self, day: int, month: str, year: int) -> None:
        """Initialize a day on the calendar with day, month and year, and no events.
        ""
        >>> d = Day(30, 'November', 2017)
        >>> d.day
        30
        >>> d.month
        'November'
        >>> d.year
        2017
        >>> d.events
        []
        ""
        
    def schedule_event(self, new_event: "Event") -> None:
        """Schedule new_event on this day, even if it overlaps with an existing event. Later we will improve this method.
        ""
        >>> d = Day(1, 'December', 2017)
        >>> e = event.Event(11, 12, 'Meeting')
        >>> d.schedule_event(e)
        >>> d.events[0] == e
        True
        """
```
def __str__(self) -> str:
    """Return a string representation of this day.
    >>> d = Day(5, 'December', 2017)
    >>> d.schedule_event(event.Event(15, 16, 'Submit A3 work'))
    >>> d.schedule_event(event.Event(16, 23, 'Celebrate!'))
    >>> print(d)
    5 December 2017:
    - Submit A3 work: from 15 to 16
    - Celebrate!: from 16 to 23
    """

if __name__ == '__main__':
    # Create day 16 December 2017.
    # Add an event "Sleep in" from 0 to 11 on 16 December 2017.
    # Add an event "Brunch" from 11 to 13 on 16 December 2017.
    # Print the day 16 December 2017, including its events.