Below is a possible solution to the `schedule_event` method that we previously wrote for the `Day` class.

```python
def schedule_event(self, new_event):
    """ (Day, Event) -> bool

    Schedule new_event on this day. Return True iff
    new_event does not overlap with an existing event.
    """
    for existing_event in self.events:
        if existing_event.overlaps(new_event):
            return False
    self.events.append(new_event)
    return True
```

Complete the `schedule_multiple_events` function below according to its docstring description. Call the `schedule_event` method as part of your solution.

```python
def schedule_multiple_events(self, event_list):
    """ (Day, list of Event) -> int

    Return the number of events in event_list that were successfully
    scheduled on this day, without overlapping with existing events.
    """
    >>> d = Day(5, 'December', 2015)
    >>> e1 = event.Event(12, 16, 'Studying')
    >>> d.schedule_event(e1)
    True
    >>> e2 = event.Event(17, 19, 'Dinner with A')
    >>> e3 = event.Event(11, 13, 'Lunch with B')
    >>> e4 = event.Event(9, 10, 'Gym')
    >>> d.schedule_multiple_events([e2, e3, e4])
    2
    """
```