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: "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: List["Event"]) -> int:
    """Return the number of events in event_list that were successfully scheduled on this day, without overlapping with existing events."
    for existing_event in self.events:
        if existing_event.overlaps(new_event):
            return False
    self.events.append(new_event)
    return True
```

```python
>>> d = Day(7, 'December', 2017)
>>> e = event.Event(17, 23, 'Celebrate end of classes')
>>> d.schedule_event(e)
True
>>> d.schedule_event(e)
True
>>> e = event.Event(18, 20, 'Eat dinner')
>>> d.schedule_event(e)
False
""

>>> 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: List["Event"]) -> int:
    """Return the number of events in event_list that were successfully scheduled on this day, without overlapping with existing events."
    for existing_event in self.events:
        if existing_event.overlaps(new_event):
            return False
    self.events.append(new_event)
    return True
```

```python
>>> d = Day(10, 'December', 2017)
>>> e1 = event.Event(12, 16, 'Studying')
>>> d.schedule_event(e1)
True
>>> e2 = event.Event(17, 19, 'Dinner with Paul')
>>> e3 = event.Event(11, 13, 'Lunch with Jen')
>>> e4 = event.Event(9, 10, 'Gym')
>>> d.schedule_multiple_events([e2, e3, e4])
2
"""