class Event:
    """A new calendar event."""

def __init__(self, start_time, end_time, event_name):
    """(Event, int, int, str) -> NoneType

    Precondition: 0 <= start_time < end_time <= 23

    Initialize a new event that starts at start_time, ends at end_time,
    and is named event_name.

    >>> e = Event(12, 13, 'Lunch')
    >>> e.start_time
    12
    >>> e.end_time
    13
    >>> e.name
    'Lunch'
    """

def rename(self, new_name):
    """(Event, str) -> NoneType

    Change the name of this event to new_name.

    >>> e = Event(12, 13, 'Lunch')
    >>>
    >>>
    """

def duration(self):
    """(Event) -> int

    Return the duration of this event.

    >>> e = Event(10, 11, 'Lecture')
    >>> e.duration()
    1
    """
def __str__(self):
    """ (Event) -> str

    Return a string representation of this event.
    
    >>> e = Event(6, 7, 'Run')
    >>> str(e)
    'Run: from 6 to 7'
    """

def __eq__(self, other):
    """ (Event, Event) -> bool

    Return True iff this event has the same start time, end time, and name as event other.

    >>> e1 = Event(6, 7, 'Run')
    >>> e2 = Event(6, 7, 'Run')
    >>> e1 == e2
    True
    """

def overlaps(self, other):
    """ (Event, Event) -> bool

    Return True iff this event overlaps with event other.

    >>> e1 = Event(6, 7, 'Run')
    >>> e2 = Event(0, 7, 'Sleep')
    >>> e1 overlaps e2
    True
    """