

```

# Empty Array declaration
events = []
joined = []

# Variable Initialization/Declaration
num_events = int(input("Number of events? "))
print(
    "\n-----\nWhen inputting in seconds format:\n\nThe start, and end of the event in this format without brackets: [start end]. \n\nFor instance, if the earliest event starts at 6:00PM, then that shall be marked as the 0th second. Then, say for instance 6:10 would be 600 seconds after the first event. Therefore, you shall input that as 600.\n-----\n"
)
counter = 0

# Input Loop
while counter < num_events:
    # Get name of event
    name = input("What is the name of this event? ")

    # Processing input of seconds
    seconds = input("The 'time' of this event in seconds format? ")
    seconds = seconds.split(" ")
    curr_event_start = int(seconds[0])
    curr_event_end = int(seconds[1])

    # Simply append it to array

    events.append([name, curr_event_start, curr_event_end])

    # Counter
    counter += 1
    print()

# Sort the array

events = sorted(events, key=lambda x: x[2])

# Processing Loop

while events:
    # Since array is already sorted, we simply need to get the first element to get the one with the earliest end time.
    curr_min = events[0]

    # This is automatically an event that we shall join.
    joined.append(curr_min)

    # Make another array to store the events that are not yet omitted. Although this might cause a Memory problem with very large inputs, I do not have enough brain power to think of something better.
    # If I do it in the array itself the indexes may overflow and it simply would just be many more weird processing stuff i guess
    new_events_array = []

    # Loop through the events to see which ones to omit and which ones to not
    for i in range(len(events)):
        # Although there is a much more efficient and more flexible and simple method, I decided to just literally check whether the events collide with each other. Besides, I think sets are quite fast.
        if len(
            set(range(curr_min[1], curr_min[2])).intersection(
                set(range(events[i][1], events[i][2]))
            )) == 0:
            # If it doesn't well, collide, don't omit it just yet
            new_events_array.append(events[i])
        # The new events array is now only filled with non-omitted stuff
    events = new_events_array

# Output it quite nicely ¯
print("-----")
print("You should join the following events:")
for i in joined:
    print(f"\n- Event name: {i[0]}")
    print(
        f"  This event shall happen between (in seconds format): {i[1]} until {i[2]}"
    )
print("\n")

```