def greedy\_interval\_scheduling(tasks):

# Sort tasks by their end times

tasks.sort(key=lambda x: x[1])

# Initialize the solution list

solution = list()

# Initially, set the finish time to the start time of the first task

finish\_time = tasks[0][0]

# Greedily select non-overlapping tasks

for task in tasks:

start, end = task

# If the current task can be scheduled without overlapping

if start >= finish\_time:

solution.append(task)

finish\_time = end

return solution

# Main Code

# each task is defined as a tuple (start time, end time)

tasks = list()

#taking user input for all the tasks

for \_ in range(int(input("Enter the number of tasks: "))):

start = int(input("Enter starting time: "))

end = int(input("Enter ending time: "))

tasks.append((start, end))

print()

#calling our function and printing the optimal solution

optimal\_solution = greedy\_interval\_scheduling(tasks)

print("Optimal Solution:", optimal\_solution)