Use a queue to simulate Round Robin Scheduling.

from queue import Queue

def simulate\_round\_robin(processes, time\_quantum):

queue = Queue()

current\_time = 0

# Add processes to the queue

for process in processes:

queue.put(process)

# Process the queue

while not queue.empty():

process = queue.get()

# Process the current process for the time quantum or until completion

if process["burst\_time"] <= time\_quantum:

current\_time += process["burst\_time"]

print("Process", process["id"], "completed at time", current\_time)

else:

current\_time += time\_quantum

process["burst\_time"] -= time\_quantum

print("Process", process["id"], "processed for", time\_quantum, "ms")

# Check if the process has completed or not

if process["burst\_time"] > 0:

queue.put(process)

# Example usage

processes = [

{"id": 1, "burst\_time": 20},

{"id": 2, "burst\_time": 15},

{"id": 3, "burst\_time": 10},

{"id": 4, "burst\_time": 5}

]

time\_quantum = 10

simulate\_round\_robin(processes, time\_quantum)