Q. FCFS Scheduling Algorithm

```
def find_waiting_time(processes, n, bt, wt, at):
    # Waiting time for first process is 0
    wt[0] = 0
    # Calculating waiting time
    for i in range(1, n):
        wt[i] = bt[i-1] + wt[i-1] - (at[i] - at[i-1])
        # If waiting time is negative, set it to zero
        if wt[i] < 0:
            wt[i] = 0
# Function to calculate turnaround time
def find_turnaround_time(processes, n, bt, wt, tat):
    for i in range(n):
        tat[i] = bt[i] + wt[i]
# Function to calculate average time
def find_avg_time(processes, n, bt, at):
    wt = [0] * n
    tat = [0] * n
    # Function calls
    find_waiting_time(processes, n, bt, wt, at)
    find_turnaround_time(processes, n, bt, wt, tat)
    # Display processes along with all details
    print("Process\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time")
    total_wt = 0
    total_tat = 0
    for i in range(n):
        total_wt += wt[i]
        total_tat += tat[i]
        print(f"{processes[i]}\t\t{at[i]}\t\t{bt[i]}\t\t{wt[i]}\t\t{tat[i]}")
    print(f"\nAverage waiting time = {total_wt / n:.2f}")
    print(f"Average turnaround time = {total_tat / n:.2f}")
```

```
# Driver code
if __name__ == "__main__":
    n = int(input("Enter number of processes: "))
    processes = []
    arrival_time = []
    burst_time = []
    for i in range(n):
        processes.append(f"P{i+1}")
        at = int(input(f"Enter arrival time of process {i+1}: "))
        bt = int(input(f"Enter burst time of process {i+1}: "))
        arrival_time.append(at)
        burst_time.append(bt)

    # Sort processes by arrival time
    processes, arrival_time, burst_time = zip(*sorted(zip(processes, arrival_time, burst_time), key=lambda x: x[1]))
    find_avg_time(processes, n, burst_time, arrival_time)
```