


```

59         System.out.println("Invalid choice. Please enter a valid option.");
60     }
61 }
62 }
63 public static void runAlgorithm(CPUScheduler scheduler) {
64     addProcesses(scheduler);
65     scheduler.process();
66     display(scheduler);
67 }
68 public static void runRoundRobin() {
69     CPUScheduler rr = new RoundRobin();
70     addProcesses(rr);
71     System.out.println("Enter time quantum for Round Robin: ");
72     Scanner scanner = new Scanner(System.in);
73     int timeQuantum = scanner.nextInt();
74     rr.setTimeQuantum(timeQuantum);
75     rr.process();
76     display(rr);
77 }
78 public static void addProcesses(CPUScheduler scheduler) {
79     Scanner scanner = new Scanner(System.in);
80     System.out.println("Enter number of processes: ");
81     int numProcesses = scanner.nextInt();
82     scanner.nextLine(); // Consume newline
83     for (int i = 0; i < numProcesses; i++) {
84         System.out.println("Enter details for process " + (i + 1) + ":");
85         System.out.print("Process Name: ");
86         String processName = scanner.nextLine();
87         System.out.print("Arrival Time: ");
88         int arrivalTime = scanner.nextInt();
89         System.out.print("Burst Time: ");
90         int burstTime = scanner.nextInt();
91         scanner.nextLine(); // Consume newline
92         scheduler.add(new Row(processName, arrivalTime, burstTime));
93     }
94 }
95 public static void display(CPUScheduler object) {
96     System.out.println("Process\tAT\tBT\tWT\tTAT");
97     for (Row row : object.getRows()) {
98         System.out.println(row.getProcessName() + "\t" + row.getArrivalTime() + "\t"
99             + row.getBurstTime() + "\t" + row.getWaitingTime() + "\t" + row.getTurnaroundTime());
100     }
101     System.out.println();
102     for (int i = 0; i < object.getTimeline().size(); i++) {
103         List<Event> timeline = object.getTimeline();
104         System.out.print(timeline.get(i).getStartTime() + "(" + timeline.get(i).getProcessName() + ")");
105         if (i == object.getTimeline().size() - 1) {
106             System.out.print(timeline.get(i).getFinishTime());
107         }
108     }
109     System.out.println("\n\nAverage WT: " + object.getAverageWaitingTime() + "\nAverage TAT: "
110         + object.getAverageTurnAroundTime());
111 }
112 }

```