2a.

Enter no of processes:

5

Process 0: AT = 2, BT = 2

Process 1: AT = 5, BT = 6

Process 2: AT = 0, BT = 4

Process 3: AT = 0, BT = 7

Process 4: AT = 7, BT = 4

Step-by-Step Execution:

1. **Initialization and Input:**
   - The program first initializes arrays to store the arrival time (a), burst time (b), process numbers (no), waiting time (wt), and turnaround time (ta).
   - After taking the number of processes (n = 5), it inputs the arrival time and burst time for each process.
2. **Sorting by Arrival Time:**
   - The processes are sorted based on their arrival times (AT). After sorting, the processes are ordered as follows:

Process	**Arrival Time (AT)**	**Burst Time (BT)**
P2	0	4
P3	0	7
P0	2	2
P1	5	6
P4	7	4

3. **Calculating Waiting Time (WT) and Turnaround Time (TAT):**
   - Now, the program calculates the start burst time (sb) for each process and from that computes the waiting time (WT) and turnaround time (TAT). The formulas are:
     - WT = Start Burst Time (sb) - Arrival Time (AT)
     - TAT = Waiting Time (WT) + Burst Time (BT)

- Let’s compute these values for each process in sequence:

For P2 (AT = 0, BT = 4):

- Start Burst Time (sb) = 0 (since it arrives first)
- WT = 0 - 0 = 0
- TAT = 0 + 4 = 4

For P3 (AT = 0, BT = 7):

- Start Burst Time (sb) = 4 (previous process burst time)
- WT = 4 - 0 = 4
- TAT = 4 + 7 = 11

For P0 (AT = 2, BT = 2):

- Start Burst Time (sb) = 11 (previous process burst time)
- WT = 11 - 2 = 9
- TAT = 9 + 2 = 11

For P1 (AT = 5, BT = 6):

- Start Burst Time (sb) = 13 (previous process burst time)
- WT = 13 - 5 = 8
- TAT = 8 + 6 = 14

For P4 (AT = 7, BT = 4):

- Start Burst Time (sb) = 19 (previous process burst time)
- WT = 19 - 7 = 12
- TAT = 12 + 4 = 16
- The computed values are:

Process	Arrival Time (AT)	Burst Time (BT)	Waiting Time (WT)	Turnaround Time (TAT)
P2	0	4	0	4
P3	0	7	4	11
P0	2	2	9	11
P1	5	6	8	14

Process	Arrival Time (AT)	Burst Time (BT)	Waiting Time (WT)	Turnaround Time (TAT)
P4	7	4	12	16

2b. Enter the no of processes: 5

Process 0: AT = 2, BT = 6

Process 1: AT = 5, BT = 2

Process 2: AT = 1, BT = 8

Process 3: AT = 0, BT = 3

Process 4: AT = 4, BT = 4

Step-by-Step Execution:

1. **Initialization and Input:**
   - The program first initializes the arrays burst_time and at to store the burst time (BT) and arrival time (AT) for each process.
   - It also calculates the total burst time, which is the sum of the burst times for all processes.
   - From the input:
     - Burst Times (BT) = [6, 2, 8, 3, 4]
     - Arrival Times (AT) = [2, 5, 1, 0, 4]
   - The total burst time is: Total Burst Time=6+2+8+3+4=23
2. **Simulating the Shortest Job Next (SJN) Scheduling:**
   - The algorithm proceeds through time (from 0 to the sum of burst times) and selects the process with the shortest burst time that has arrived (i.e., its arrival time is less than or equal to the current time).

- At each time step, the process with the smallest burst time is selected and executed until it completes, and its burst time is set to 0.

3. **Step-by-Step Process Execution:**
   - **Time 0-3:**
     - At time = 0, processes that have arrived are P3 (AT = 0, BT = 3).
     - P3 is the smallest, so it runs from time = 0 to time = 3.
     - **WT (Waiting Time) for P3 = 0** (since it starts immediately after arrival).
     - **TAT (Turnaround Time) for P3 = 3** (it finishes at time = 3).
   - **Time 3-9:**
     - At time = 3, processes that have arrived are P0 (AT = 2, BT = 6), P2 (AT = 1, BT = 8), and P4 (AT = 4, BT = 4).
     - P0 is the smallest, so it runs from time = 3 to time = 9.
     - **WT for P0 = 3 - 2 = 1** (starts after waiting from AT = 2).
     - **TAT for P0 = 9 - 2 = 7** (finishes at time = 9).
   - **Time 9-11:**
     - At time = 9, processes that have arrived are P1 (AT = 5, BT = 2) and P4 (AT = 4, BT = 4).
     - P1 is the smallest, so it runs from time = 9 to time = 11.
     - **WT for P1 = 9 - 5 = 4** (starts after waiting from AT = 5).
     - **TAT for P1 = 11 - 5 = 6** (finishes at time = 11).
   - **Time 11-15:**
     - At time = 11, the only process left is P4 (AT = 4, BT = 4).
     - P4 runs from time = 11 to time = 15.
     - **WT for P4 = 11 - 4 = 7** (starts after waiting from AT = 4).
     - **TAT for P4 = 15 - 4 = 11** (finishes at time = 15).
   - **Time 15-23:**
     - At time = 15, the only process left is P2 (AT = 1, BT = 8).
     - P2 runs from time = 15 to time = 23.
     - **WT for P2 = 15 - 1 = 14** (starts after waiting from AT = 1).
     - **TAT for P2 = 23 - 1 = 22** (finishes at time = 23).

4. **Calculating Waiting Time (WT) and Turnaround Time (TAT):** After calculating the individual WT and TAT for each process, we summarize the results:

Process	**AT**	**BT**	**WT**	**TAT**
P3	0	3	0	3
P0	2	6	1	7
P1	5	2	4	6
P4	4	4	7	11
P2	1	8	14	22

2c. Enter Total Number of Processes: 3

Process 1: Arrival Time = 0, Burst Time = 10

Process 2: Arrival Time = 1, Burst Time = 8

Process 3: Arrival Time = 2, Burst Time = 7

Enter Time quantum: 5

Step-by-Step Execution:

1. **Initialization and Input:**
   - The program takes the number of processes n = 3 and stores the arrival time and burst time for each process.
   - For this input:
     - Arrival Times (AT) = [0, 1, 2]
     - Burst Times (BT) = [10, 8, 7]
     - Temporary Burst Times (temp_burst_time) = [10, 8, 7] (to track remaining time)

The time quantum is 5, meaning each process gets up to 5 units of time in each round.

2. **Simulating Round Robin Scheduling:**
   - The simulation runs until all processes have completed their burst time.
   - At each time step, the process executes for the lesser of its remaining burst time or the time quantum.
3. **Step-by-Step Process Execution:**
   - **First Cycle (Time 0 to 15):**
     - **P1** runs from time 0 to time 5 (uses full quantum). Remaining burst time = 5.
     - **P2** runs from time 5 to time 10 (uses full quantum). Remaining burst time = 3.
     - **P3** runs from time 10 to time 15 (uses full quantum). Remaining burst time = 2.
   - **Second Cycle (Time 15 to 25):**
     - **P1** runs from time 15 to time 20 (uses full quantum). Remaining burst time = 0 (completes).
       - **Turnaround Time (TAT)** for P1 = 20 - 0 = 20
       - **Waiting Time (WT)** for P1 = 20 - 10 = 10
     - **P2** runs from time 20 to time 23 (remaining 3 units). Remaining burst time = 0 (completes).
       - **TAT** for P2 = 22 - 1 = 22
       - **WT** for P2 = 22 - 8 = 14
     - **P3** runs from time 23 to time 25 (remaining 2 units). Remaining burst time = 0 (completes).
       - **TAT** for P3 = 23 - 2 = 23
       - **WT** for P3 = 23 - 7 = 16
4. **Calculating Waiting Time (WT) and Turnaround Time (TAT):** After calculating the individual WT and TAT for each process, we summarize the results:

Process	**BT**	**TAT**	**WT**
P1	10	20	10
P2	8	22	14
P3	7	23	16

2d. Enter Total Number of Processes: 7

Details for each process:

Process A: Arrival Time = 0, Burst Time = 3, Priority = 2

Process B: Arrival Time = 2, Burst Time = 5, Priority = 6

Process C: Arrival Time = 1, Burst Time = 4, Priority = 3

Process D: Arrival Time = 4, Burst Time = 2, Priority = 5

Process E: Arrival Time = 6, Burst Time = 9, Priority = 7

Process F: Arrival Time = 5, Burst Time = 4, Priority = 4

Process G: Arrival Time = 7, Burst Time = 10, Priority = 10

Execution Process:

1. **Arrival Time Sorting:** The processes are initially sorted based on their arrival times, which results in this order:
    - ○ **Order after sorting by Arrival Time:**

Process Name	Arrival Time	Burst Time	Priority
A	0	3	2
C	1	4	3
B	2	5	6
D	4	2	5
F	5	4	4
E	6	9	7
G	7	10	10

 Executing Processes Based on Priority: At each point in time, the process with the highest priority (smallest priority value) that has already arrived is selected for execution.

 Step-by-Step Execution:

- **Time 0:** Process **A** is selected (priority 2). It runs for 3 units of time.
  Completion Time (CT) = 3, Waiting Time (WT) = 0, Turnaround Time (TAT) = 3 - 0 = 3.

- **Time 3:** Process **C** is selected (priority 3). It runs for 4 units of time.
  Completion Time (CT) = 7, Waiting Time (WT) = 7 - 1 - 4 = 2, Turnaround Time (TAT) = 7 - 1 = 6.
- **Time 7:** Process **F** is selected (priority 4). It runs for 4 units of time.
  Completion Time (CT) = 11, Waiting Time (WT) = 11 - 5 - 4 = 2, Turnaround Time (TAT) = 11 - 5 = 6.
- **Time 11:** Process **D** is selected (priority 5). It runs for 2 units of time.
  Completion Time (CT) = 13, Waiting Time (WT) = 13 - 4 - 2 = 7, Turnaround Time (TAT) = 13 - 4 = 9.
- **Time 13:** Process **B** is selected (priority 6). It runs for 5 units of time.
  Completion Time (CT) = 18, Waiting Time (WT) = 18 - 2 - 5 = 11, Turnaround Time (TAT) = 18 - 2 = 16.
- **Time 18:** Process **E** is selected (priority 7). It runs for 9 units of time.
  Completion Time (CT) = 27, Waiting Time (WT) = 27 - 6 - 9 = 12, Turnaround Time (TAT) = 27 - 6 = 21.
- **Time 27:** Process **G** is selected (priority 10). It runs for 10 units of time.
  Completion Time (CT) = 37, Waiting Time (WT) = 37 - 7 - 10 = 20, Turnaround Time (TAT) = 37 - 7 = 30.

Final Table:

Process	**AT**	**BT**	**Priority**	**WT**	**TAT**
A	0	3	2	0	3
C	1	4	3	2	6
F	5	4	4	2	6
D	4	2	5	7	9
B	2	5	6	11	16
E	6	9	7	12	21
G	7	10	10	20	30