Question NO 01:-

Reason:

For a system with 4 cores and a 1-to-1 threading model, I create 4 threads - one for each core - to maximize efficiency and for utilizing all resources efficiently.

Question NO 02:-

Answer:-

With 8 cores available and using a 1 to 1 threading model, its optimal to create 8 threads for the task.

Reasons:

→ Max CPU Utilization:-

Utilizing all 8 cores with individual threads ensures that each core is fully engaged, maximizing computational thruput

➡ Efficient Resource Utilization:-

Since, each thread is can run independently on its core, there minimal overhead from context switching, ensuring efficient use of CPU resources.

➡ Balanced Workload:-

By distributing the workload across 8 threads, the task of determining number of frequency of specific number in the array can be performed concurrently and efficiently.

➡ Therefore, for a system with 8 cores and a 1-to-1 threading model, using 8 threads ensures performance and efficiently utilizing available computational resources.★

Question No. 03:- *

Answer.

➡ I create two threads because the number of cores in system are 2 and its the optimal condition for a system with two cores as it maximize parallelism and ensure all availble processing power is used.*

Reasons:

With 2 core available and 1-to-1 threading model:

Optimal thread count is 2.
Each core handle one thread efficiently.
Minimizes context switching overhead.
Maintains efficient resource utilization.
Balances workload distribution effectively.

: Hassan Zaib Jadoon
22PWCSE2144

⇒ ANSWER # 4:-

(a) Gantt Charts:

→ First Come, First Serve:-*

| P1 | P2 | P3 | P4 | P5 |
0 5 8 9 16 20

→ Shortest Job First:-*

| P3 | P2 | P5 | P1 | P4 |
0 1 4 8 13 20

→ Shortest Remaining Time:-*

| P3 | P2 | P5 | P1 | P4 |
0 1 4 8 13 20

→ Non-Preemtive [redacted]:-*

| P2 | P5 | P3 | P4 | P2 |
0 5 9 10 17 20

→ Preemptive Priority:-

| P1 | P5 | P3 | P4 | P2 |
2 5 9 10 13 20

TAT of Each Process:-

→ First Come, First Serve:-

Process	Burst Time	Waiting Time	TAT = WT + BT
P_1	5	0	5
P_2	3	5	8
P_3	1	8	9
P_4	7	9	16
P_5	4	16	20
		7.6	11.6

→ Shortest Job First:-

	BT	WT	TAT
P_3	1	0	1
P_2	3	1	4
P_5	4	4	8
P_1	5	8	13
P_4	7	13	20
		5.2	9.2

Shortest Remaining Time:-

	BT	WT	TAT
P_3	1	0	0
P_2	3	1	5
P_5	4	4	9
P_1	5	8	10
P_4	7	13	17
		5.2	8.2

Premptive Property:-

	Priority	TAT	WT
P_1	4	5	0
P_5	3	9	5
P_3	2	10	9
P_4	2	17	10
P_2	1	20	17
		12.2	8.2

CPU Utilization Rate:-

As we know that C.U.R is Total Required time by total time taken into 100,

$$C.U.R = \frac{\text{Total Required Time}}{\text{Total Time Taken}} \times 100.$$

(i) Shortest Job First:-

$20/20 \times 100 \Rightarrow 100\%$

(ii) Shortest Remaining Time:-

$20/20 \times 100\% \Rightarrow 100$

(iii) Non-Premptive Priority:-

$20/20 \times 100\% \Rightarrow 100\%$.

(iv) First Come First Serve:-

$20/20 \times 100\% \Rightarrow 100\%$

(v) Premptive Priority:-

$20/20 \times 100\% \Rightarrow 100$

Waiting Time for Each Process:-

The waiting time for each process has already found in part b.

Best Performance:-

SJF and SRT have the best performance because of they have less average TAT and WT than the other one.

Round Robin SA:-

P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_4	P_5	P_1	P_4
0	2	4	5	7	9	11	12	14	16	17	20

→ T.A.T and W.T:-

	TAT	W.T
P_1	17	12
P_2	12-2 ~~=10~~	9
P_3	5-4 ⇒ 1	4
P_4	20-5 ⇒ 5	13
P_5	16-7=9	12

➡ CPU Utilization Rate:-

$20/20 \times 100\% \Rightarrow 100\%$

➡ Question No 05:-

➡: ANSWER:

(a): Gantt Chart:

P_1	P_2	P_3	P_4	P_3	P_5	P_4	P_6	P_2	P_3	P_4
0	15	20	30	40	45	50	55	70	85	90	95

(b): Turn Around Time:

Process	Priority	BT	TAT	W.T = T.AT - BT.
P_1	8	15	15	0
P_2	3	20	85	65
P_3	4	20	70	50
P_4	4	20	70	50
P_5	5	5	5	0
P_6	9	15	15	0

Waiting Time:

Process	Priority	BT	TAT	WT = TAT - BT
P_1	8	15	15	0
P_2	3	20	85	65
P_3	4	20	70	50
P_4	4	20	70	50
P_5	5	5	5	0
P_6	5	15	15	0