CPU SCHEDULING ALGORITHMS

4. Shortest Remaining Time

```
// Disk Scheduling - SRT
#include<stdio.h>
struct process{
	int at,bt,rt;
}p[10];
main()
{
	int endTime,i,smallest;
	int remain=0,n,time,sum_wait=0,sum_turnaround=0;
	printf("\n Enter the Number of Processes : ");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("Process P[%d] : \n",i+1);
		printf("\n Arrival Time : ",i+1);
		scanf("%d",&p[i].at);
		printf("\n Burst Time : ",i+1);
		scanf("%d",&p[i].bt);
		p[i].rt = p[i].bt;
	}
	printf("\n\nProcess\tTurnaround Time| Response Time\n\n");
	p[9].rt=9999;
	for(time=0;remain!=n;time++)
	{
		smallest=9;
		for(i=0;i<n;i++)
		{
			if(p[i].at<=time && p[i].rt<p[smallest].rt && p[i].rt>0)
			{
				smallest=i;
			}
		}
		p[smallest].rt--;
		if(p[smallest].rt==0)
		{
			remain++;
			endTime=time+1;
			printf("\nP[%d]\t|\t%d\t|\t%d",smallest+1,endTime-p[smallest].at,endTime-
p[smallest].bt-p[smallest].at);
			sum_wait+=endTime-p[smallest].bt-p[smallest].at;
			sum_turnaround+=endTime-p[smallest].at;
		}
	}
	printf("\n\nAverage waiting time = %f\n",sum_wait*1.0/n);
	printf("Average Turnaround time = %f\n\n",sum_turnaround*1.0/5);
}
```

CPU SCHEDULING ALGORITHMS
4. Shortest Remaining Time

OUTPUT SRT :

Enter the Number of Processes : 5
Process P[1] :

Arrival Time : 0

Burst Time : 3
Process P[2] :

Arrival Time : 2

Burst Time : 6
Process P[3] :

Arrival Time : 4

Burst Time : 4
Process P[4] :

Arrival Time : 6

Burst Time : 5
Process P[5] :

Arrival Time : 8

Burst Time : 2

Process |Turnaround Time| Response Time

P[1] | 3 | 0
P[3] | 4 | 0
P[5] | 2 | 0
P[2] | 13 | 7
P[4] | 14 | 9

Average waiting time = 3.200000
Average Turnaround time = 7.200000