#FCFS

```
#include<stdio.h>
typedef struct process{
	int pid; //process id
	int at;		//arrival time
	int st;		//burst time
	int tat; //turn around time
	int wt;		//waiting time
}p;
int main(){
	int n;
	printf("Enter no. of process");
	scanf("%d",&n);
	p process[n];
	int i,j;
	p swap;
	int time;
	for(i = 0; i < n; i++){
		process[i].pid = i;
		printf("Process %d arrival time ",i);
		scanf("%d",&process[i].at);
		printf("Process %d burst time   ",i);
		scanf("%d",&process[i].st);
	}
	for(i = 0;i < n - 1;i++){
		for(j = 0; j < n-i-1; j++){
			if (process[j].at > process[j+1].at) {
			swap = process[j];
			process[j]   = process[j+1];
				process[j+1] = swap;
			}
		}
	}
	for(i = 0; i < n; i++){
		printf("Process   :%d\t",process[i].pid);
	}
	time = 0;
	printf("\n");
```

```
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("Time start:%d\t", time);
	}
	time = 0;
	printf("\n\n\n");
	printf("pid\tat\tst\ttat\twt\n");
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("%d\t%d\t%d\t%d\t%d\n",process[i].pid,process[i].at,process[i].st,time - process[i].at,time - process[i].at - process[i].st);
	}
}
```

#Priority Scheduling

```
#include<stdio.h>
typedef struct process{
	int pid; //process id
	int at;		//arrival time
	int st;		//burst time
	int tat; //turn around time
	int wt;		//waiting time
	int priority;
}p;
int main(){
	int n;
	printf("Enter no. of process");
	scanf("%d",&n);
	p process[n];
	int i,j;
	p swap;
	int time = 0;
	for(i = 0; i < n; i++){
		process[i].pid = i;
		printf("Process %d arrival time  ",i);
		scanf("%d",&process[i].at);
		printf("Process %d burst time    ",i);
		scanf("%d",&process[i].st);
		printf("Process %d priority      ",i);
		scanf("%d",&process[i].priority);
	}
	for(i = 0;i < n - 1;i++){
		for(j = i + 1; j < n; j++){
			if ((process[i].priority > process[j].priority) && (process[j].at <= time) ) {
			swap = process[i];
			process[i]   = process[j];
				process[j] = swap;
			}
		}
		time = time + process[i].st;
	}
```

```
	printf("\n\n\n");
	for(i = 0; i < n; i++){
		printf("Process   :%d\t",process[i].pid);
	}
	time = 0;
	printf("\n");
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("Time start:%d\t", time);
	}
	time = 0;
	printf("\n\n\n");
	printf("pid\tat\tst\ttat\twt\n");
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("%d\t%d\t%d\t%d\t%d\n",process[i].pid,process[i].at,process[i].st,time -
process[i].at,time - process[i].at - process[i].st);
	}
}
```

#Round Robin

```
#include<stdio.h>
typedef struct process{
	int pid; //process id
	int at;		//arrival time
	int st;		//burst time
	int tat; //turn around time
	int wt;		//waiting time
}p;
int main(){
	int n;
	printf("Enter no. of process  ");
	scanf("%d",&n);
	p process[n];
	int i,j;
	p swap;
	int time = 0;
	int quantum;
	for(i = 0; i < n; i++){
		process[i].pid = i;
		printf("Process %d arrival time   ",i);
		scanf("%d",&process[i].at);
		printf("Process %d burst time     ",i);
		scanf("%d",&process[i].st);
	}
	printf("\nEnter value of quantum   ");
	scanf("%d",&quantum);
	for(i = 0; i < n; i++){
		time = time + process[i].st;
	}
	int t;
	printf("\nProcess  End Time\n");
	while(time != t){
		for(i = 0;i < n;i++){
			if(process[i].st >=  quantum){
				process[i].st = process[i].st - quantum;
				t = t + quantum;
```

```
                    printf("%d\t%d\n",process[i].pid,t);
                }
                else if(process[i].st != 0){
                    t = t + process[i].st;
                    process[i].st = 0;
                    printf("%d\t%d\n",process[i].pid,t);
                }
            }
        }
}
```

#SRTF

```
#include<stdio.h>
typedef struct process{
	int pid; //process id
	int at;		//arrival time
	int st;		//burst time
	int tat; //turn around time
	int wt;		//waiting time
}p;
int main(){
	int n;
	printf("Enter no. of process");
	scanf("%d",&n);
	p process[n];
	int i,j;
	p swap;
	int time = 0;
	for(i = 0; i < n; i++){
		process[i].pid = i;
		printf("Process %d arrival time",i);
		scanf("%d",&process[i].at);
		printf("Process %d burst time",i);
		scanf("%d",&process[i].st);
	}
	for(i = 0; i < n; i++){
		time = time + process[i].st;
	}
	int min = 100;
	int t;
	for(i = 0; i < time;i++){
		min  = 100;
		for(j = 0; j < n;j++){
			if(process[j].at <= i){
				if((process[j].st < min ) && (process[j].st > 0 )){
					min = process[j].st;
					t = j;
				}
```

```
                }
            }
            process[t].st = process[t].st - 1;
            printf("%d\n",process[t].pid);
        }
}
```

#SJF

```
#include<stdio.h>
typedef struct process{
	int pid; //process id
	int at;		//arrival time
	int st;		//burst time
	int tat;	//turn around time
	int wt;		//waiting time
}p;
int main(){
	int n;
	printf("Enter no. of process   ");
	scanf("%d",&n);
	p process[n];
	int i,j;
	p swap;
	int time = 0;
	for(i = 0; i < n; i++){
		process[i].pid = i;
		printf("Process %d arrival time   ",i);
		scanf("%d",&process[i].at);
		printf("Process %d burst time     ",i);
		scanf("%d",&process[i].st);
	}
	for(i = 0;i < n - 1;i++){
		for(j = i + 1; j < n; j++){
			if ((process[i].st > process[j].st) && (process[j].at <= time) ) {
			swap = process[i];
			process[i]   = process[j];
				process[j] = swap;
			}
		}
		time = time + process[i].st;
	}
	for(i = 0; i < n; i++){
		printf("Process_id:%d\t",process[i].pid);
	}
```

```
	time = 0;
	printf("\n\n\n");
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("time start:%d\t", time);
	}
	time = 0;
	printf("\n\n\n");
	printf("pid\tat\tst\ttat\twt\n");
	for(i = 0; i < n; i++){
		time = time + process[i].st;
		printf("%d\t%d\t%d\t%d\t%d\n",process[i].pid,process[i].at,process[i].st,time -
process[i].at,time - process[i].at - process[i].st);
	}
}
```