Implementation of FCFS

```
#include<stdio.h>
#include<conio.h>
#define mx 30

int main(){
   //initialization part
   int
i,j,n,BurstTime[mx],ArrivalTime[mx],WaitingTime[mx],TurnAroundTime[mx],tmp[mx];
   float AvgWaitTime=0, AvgTurnArTime=0;

   //input part here
   printf("Enter the no of process: ");
   scanf("%d",&n);
   printf("Enter the burst time of the process: ");
   for(i=0; i<n; i++){
      scanf("%d",&BurstTime[i]);
   }
   printf("Enter the arrival time of the process: ");
   for(i=0; i<n; i++){
      scanf("%d",&ArrivalTime[i]);
   }
   tmp[0]=0;
   printf("process\t burst time\t arrival time\t waiting time\t turn around time\n");
   //Logic implementation part here
   for(i=0; i<n; i++){
      WaitingTime[i]=0;
      TurnAroundTime[i]=0;
      tmp[i+1]=tmp[i]+BurstTime[i];
      WaitingTime[i]=tmp[i]-ArrivalTime[i];
      TurnAroundTime[i]=WaitingTime[i]+BurstTime[i];
      AvgWaitTime=AvgWaitTime+WaitingTime[i];
      AvgTurnArTime=AvgTurnArTime+TurnAroundTime[i];

printf("%d\t%d\t\t%d\t\t%d\t\t%d\n",i+1,BurstTime[i],ArrivalTime[i],WaitingTime[i],TurnAro
undTime[i]);
   }
   AvgWaitTime=AvgWaitTime/n;
   AvgTurnArTime=AvgTurnArTime/n;
```

```
    printf("Average waiting time %f\n",AvgWaitTime);
    printf("Average turn around time %f\n",AvgTurnArTime);

}
```

Implementation of SJF

```
#include<stdio.h>
#include<conio.h>
#define mx 30

int main(){
    //initialization part
    int
i,j,n,BurstTime[mx],ArrivalTime[mx],WaitingTime[mx],TurnAroundTime[mx],tmp[mx];
    float AvgWaitTime=0, AvgTurnArTime=0;

    //input part here
    printf("Enter the no of process: ");
    scanf("%d",&n);
    printf("Enter the burst time of the process: ");
    for(i=0; i<n; i++){
        scanf("%d",&BurstTime[i]);
    }
    printf("Enter the arrival time of the process: ");
    for(i=0; i<n; i++){
        scanf("%d",&ArrivalTime[i]);
    }
    tmp[0]=0;
    printf("process\t burst time\t arrival time\t waiting time\t turn around time\n");
    //Logic implementation part here
    for(i=0; i<n; i++){
        WaitingTime[i]=0;
        TurnAroundTime[i]=0;
        tmp[i+1]=tmp[i]+BurstTime[i];
        WaitingTime[i]=tmp[i]-ArrivalTime[i];
        TurnAroundTime[i]=WaitingTime[i]+BurstTime[i];
        AvgWaitTime=AvgWaitTime+WaitingTime[i];
        AvgTurnArTime=AvgTurnArTime+TurnAroundTime[i];
```

```
printf("%d\t%d\t\t%d\t\t%d\t\t%d\n",i+1,BurstTime[i],ArrivalTime[i],WaitingTime[i],TurnAro
undTime[i]);
    }
    AvgWaitTime=AvgWaitTime/n;
    AvgTurnArTime=AvgTurnArTime/n;
    printf("Average waiting time %f\n",AvgWaitTime);
    printf("Average turn around time %f\n",AvgTurnArTime);

}
```

Implementation of RR

```
#include<stdio.h>
#define mx 30

int main(){
    // Initialization part
    int i, n, QuaTime, cnt = 0, tmp, sq = 0, rem_bt[mx], BurstTime[mx], WaitingTime[mx],
TurnAroundTime[mx];
    float AvgWaitTime = 0, AvgTurnArTime = 0;

    // Input part
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    printf("Enter the burst time of the processes: ");
    for(i = 0; i < n; i++){
        scanf("%d", &BurstTime[i]);
        rem_bt[i] = BurstTime[i];
    }
    printf("Enter the Quantum Time: ");
    scanf("%d", &QuaTime);

    // Processing part
    while(1){
        int done = 1;
        for(i = 0; i < n; i++){
            if(rem_bt[i] > 0) {
```

```
            done = 0;  // There is a pending process

            if(rem_bt[i] > QuaTime){
               rem_bt[i] -= QuaTime;
               sq += QuaTime;
            }
            else {
               sq += rem_bt[i];
               rem_bt[i] = 0;
               TurnAroundTime[i] = sq;
            }
         }
      }
      if(done == 1) // If all processes are done
         break;
   }

   printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");
   for(i = 0; i < n; i++){
      WaitingTime[i] = TurnAroundTime[i] - BurstTime[i];
      AvgWaitTime += WaitingTime[i];
      AvgTurnArTime += TurnAroundTime[i];
      printf("%d\t%d\t\t%d\t\t%d\n", i + 1, BurstTime[i], WaitingTime[i],
TurnAroundTime[i]);
   }

   AvgWaitTime /= n;
   AvgTurnArTime /= n;
   printf("Average waiting time: %f\n", AvgWaitTime);
   printf("Average turnaround time: %f\n", AvgTurnArTime);

   return 0;
}
```