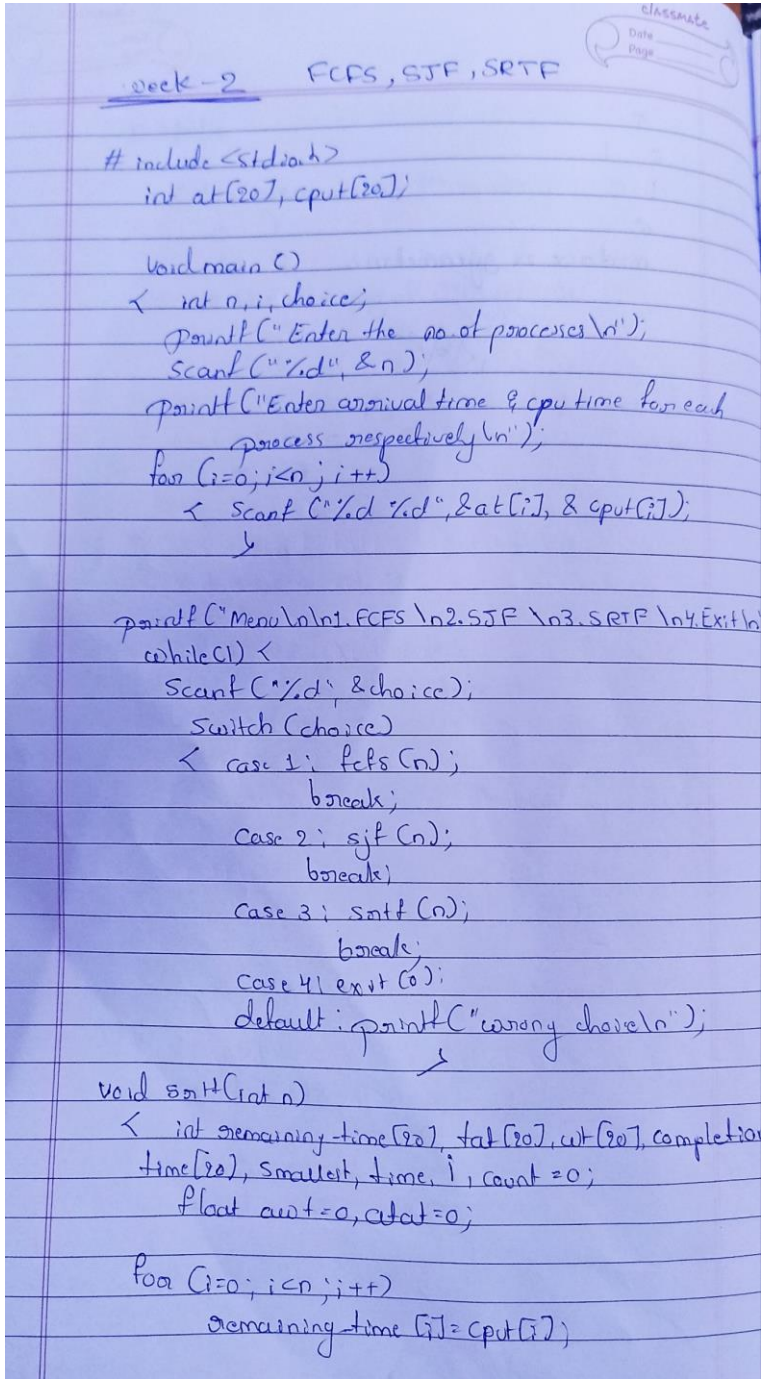
Q) WAP to execute FCFS, SJF and SRTF for process scheduling



Week-2 FCFS, SJF, SRTF

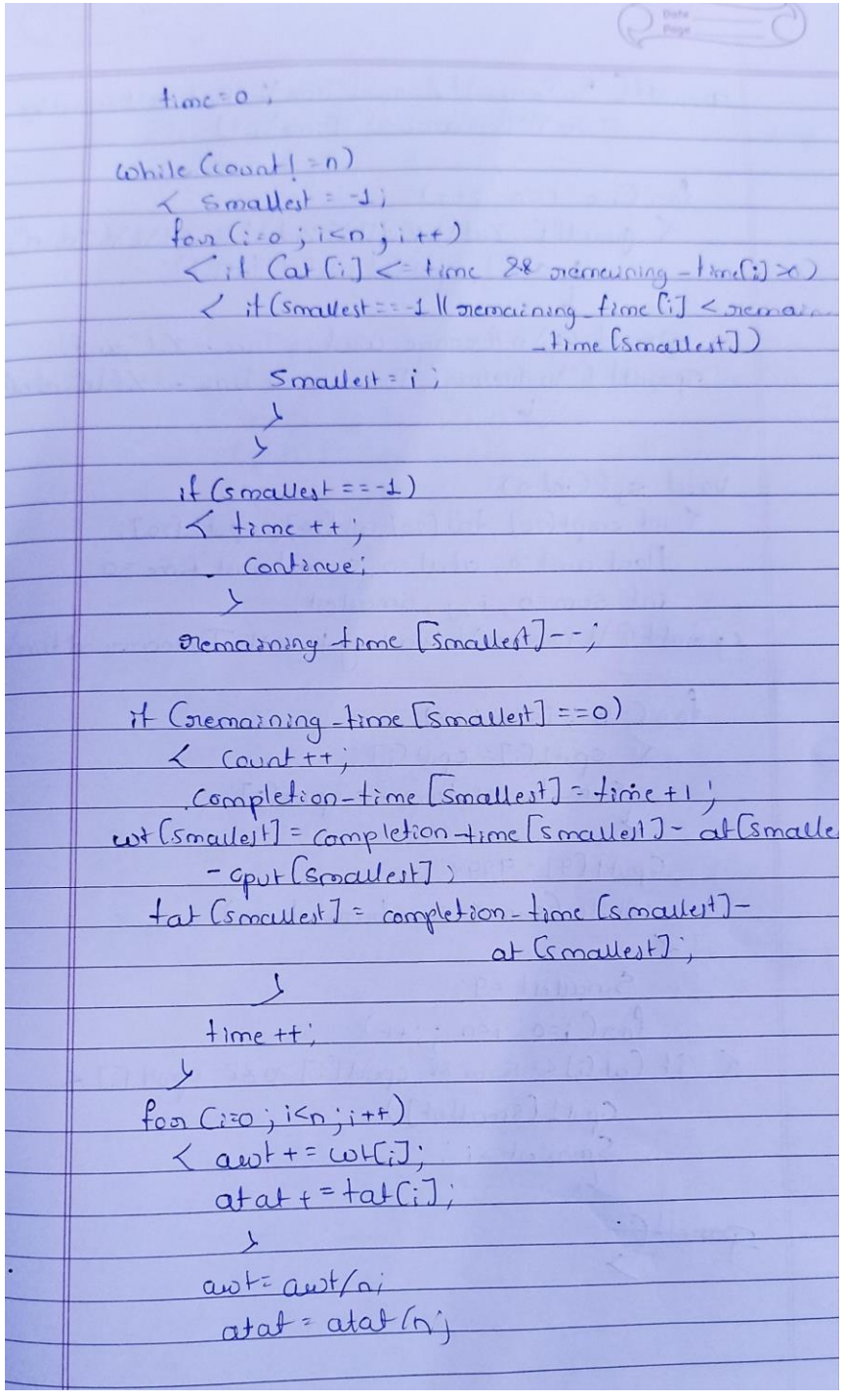
```
#include<stdio.h>
int at[20], cput[20];

void main()
{ int n, i, choice;
  printf("Enter the no of processes\n");
  scanf("%d", &n);
  printf("Enter arrival time & cpu time for each
      process respectively\n");
  for(i=0; i<n; i++)
    { scanf("%d %d", &at[i], &cput[i]);
    }

printf("Menu\n\n1. FCFS \n2. SJF \n3. SRTF \n4. Exit\n
while(1) {
  scanf("%d", &choice);
  switch(choice)
  { case 1: fcfs(n);
      break;
    case 2: sjf(n);
      break;
    case 3: srtf(n);
      break;
    case 4: exit(0);
    default: printf("wrong choice\n");
  }

void srtf(int n)
{ int remaining_time[20], tat[20], wt[20], completion
  time[20], smallest, time, i, count=0;
  float awt=0, atat=0;

for(i=0; i<n; i++)
    remaining_time[i]=cput[i];
```



```
time=0;

while(count!=n)
{ smallest=-1;
  for(i=0; i<n; i++)
  { if(at[i]<=time && remaining_time[i]>0)
    { if(smallest==-1 || remaining_time[i]<remai
        _time[smallest])
      { smallest=i;
      }
    }
  }
  if(smallest==-1)
  { time++;
    continue;
  }
  remaining_time[smallest]--;

if(remaining_time[smallest]==0)
{ count++;
  completion_time[smallest]=time+1;
wt[smallest]=completion_time[smallest]-at[smalle
    -cput[smallest];
  tat[smallest]=completion_time[smallest]-
                at[smallest];
  }
  time++;
}
for(i=0; i<n; i++)
{ awt+=wt[i];
  atat+=tat[i];
}
awt=awt/n;
atat=atat/n;
```

```
printf("\n Process\t Arrival Time\t CPU time\t Waiting
      Time\t Turnaround Time\n");

for(i=0; i<n; i++)
{ printf("%d\t %d\t\t %d\t\t %d\t\t %d\n",
     i, at[i], cput[i], wt[i], tat[i]);
}
printf("\n Average waiting Time -- %f", awt);
printf("\n Average Turnaround Time -- %f\n", atat);
}

void sjf(int n)
{ int cmpt[20], tat[20], wt[20], cput1[20];
  float awt=0, atat=0, sum_burst_time=0;
  int sum=0, i, j, smallest;
printf("\t Process\t Waiting time\t Turnaround time\n");

  for(i=0; i<n; i++)
  { cput1[i] = cput[i];
    sum_burst_time += cput[i];
  }
  cput1[9] = 9999;
  while(sum < sum_burst_time)
  {
    smallest = 9;
    for(i=0; i<n; i++)
{ if(at[i] <= sum && cput1[i] > 0 && cput1[i] <
      cput1[smallest])
      smallest = i;
  }
printf(
}
```

```
void fcfs(int n)
{ int cmpt[20], tat[20], wt[20], pname[20], temp;
  float awt=0, atat=0;
  int sum=0, i;
  for(i=0; i<n; i++)
  { pname[i] = i;
  }
  for(i=0; i<n; i++)
  { if(at[i] == at[i+1] && cput[i] > cput[i+1])
    { temp = cput[i];
      cput[i] = cput[i+1];
      cput[i+1] = temp;
      temp = pname[i];
      pname[i] = pname[i+1];
      pname[i+1] = temp;
    }
  }
  for(i=0; i<n; i++)
  { sum += cput[i];
    cmpt[i] = sum;
    tat[i] = cmpt[i] - at[i];
    wt[i] = tat[i] - cput[i];
  }
for(i=0; i<n; i++)
  { awt += wt[i];
    atat += tat[i];
  }
  awt = awt/n;
  atat = atat/n;
printf("\t Process\t Arrival time\t CPU Time\t
    Waiting time\t Turnaround time\n");
```

```
for(i=0; i<n; i++)
{ printf("\n\t%d\t\t%d\t\t%d\t\t%d\t\t%d",
    pname[i], at[i], cput[i], wt[i], tat[i]);
}
printf("\n Avg waiting time -- %f", awt);
printf("\n Avg Turnaround time -- %f\n", atat);
```

Output:

Enter the number of processes

4

Enter arrival time and cpu time for each process respectively.

0 3
1 6
4 4
6 2

Menu

1. FCFS 2. STF 3. SRTF 4. Exit

1

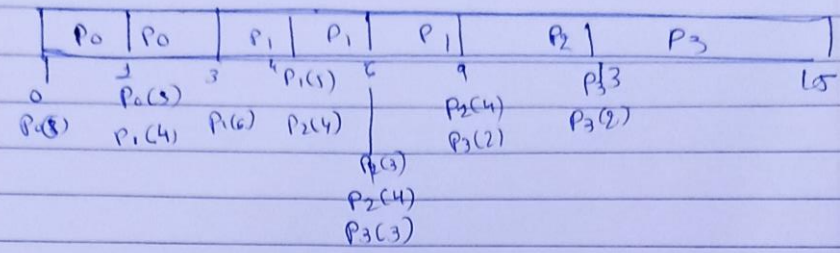
Process	Arrival time	CPU time	Waiting time	Turnaround time
P_0	0	3	0	3
P_1	1	4	2	8
P_2	4	6	5	4
P_3	6	2	7	9

Average waiting time - 3.50000

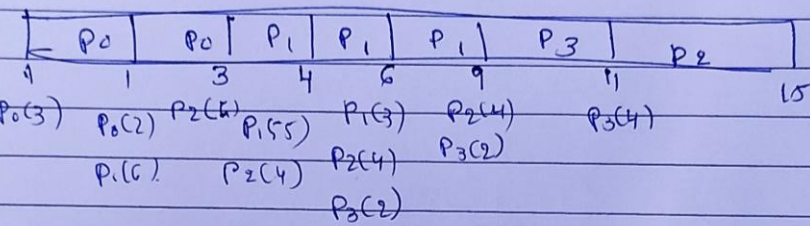
Average Turnaround time — 7.25000

FCFS

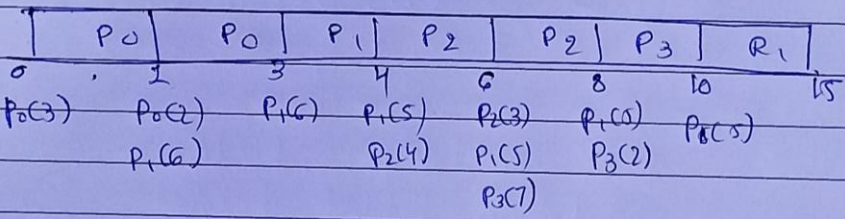
1)



2) SJF



3) SRTF



Output:

```
Enter the number of processes
4
Enter arrival time and cpu time for each process respectively
0 3
1 6
4 4
6 2
Menu

1.FCFS
2.SJF(Non Preemptive)
3.SRTF(Preemptive)
4.Exit
1
        PROCESS         ARRIVAL TIME    CPU TIME        WAITING TIME    TURNAROUND TIME

        P0               0               3              0               3
        P1               1               6              2               8
        P2               4               4              5               9
        P3               6               2              7               9
Average Waiting Time -- 3.500000
Average Turnaround Time -- 7.250000

2
        PROCESS          WAITING TIME    TURNAROUND TIME
        P[0]             3               0
        P[1]             8               2
        P[3]             5               3
        P[2]             11              7

Average Waiting Time -- 6.750000
Average Turnaround Time -- 3.000000
3
```

```
Average Waiting Time -- 6.750000
Average Turnaround Time -- 3.000000
3

Process Arrival Time    CPU Time        Waiting Time    Turnaround Time
0       0               3               0               3
1       1               6               8               14
2       4               4               0               4
3       6               2               2               4

Average Waiting Time -- 2.500000
Average Turnaround Time -- 6.250000
```