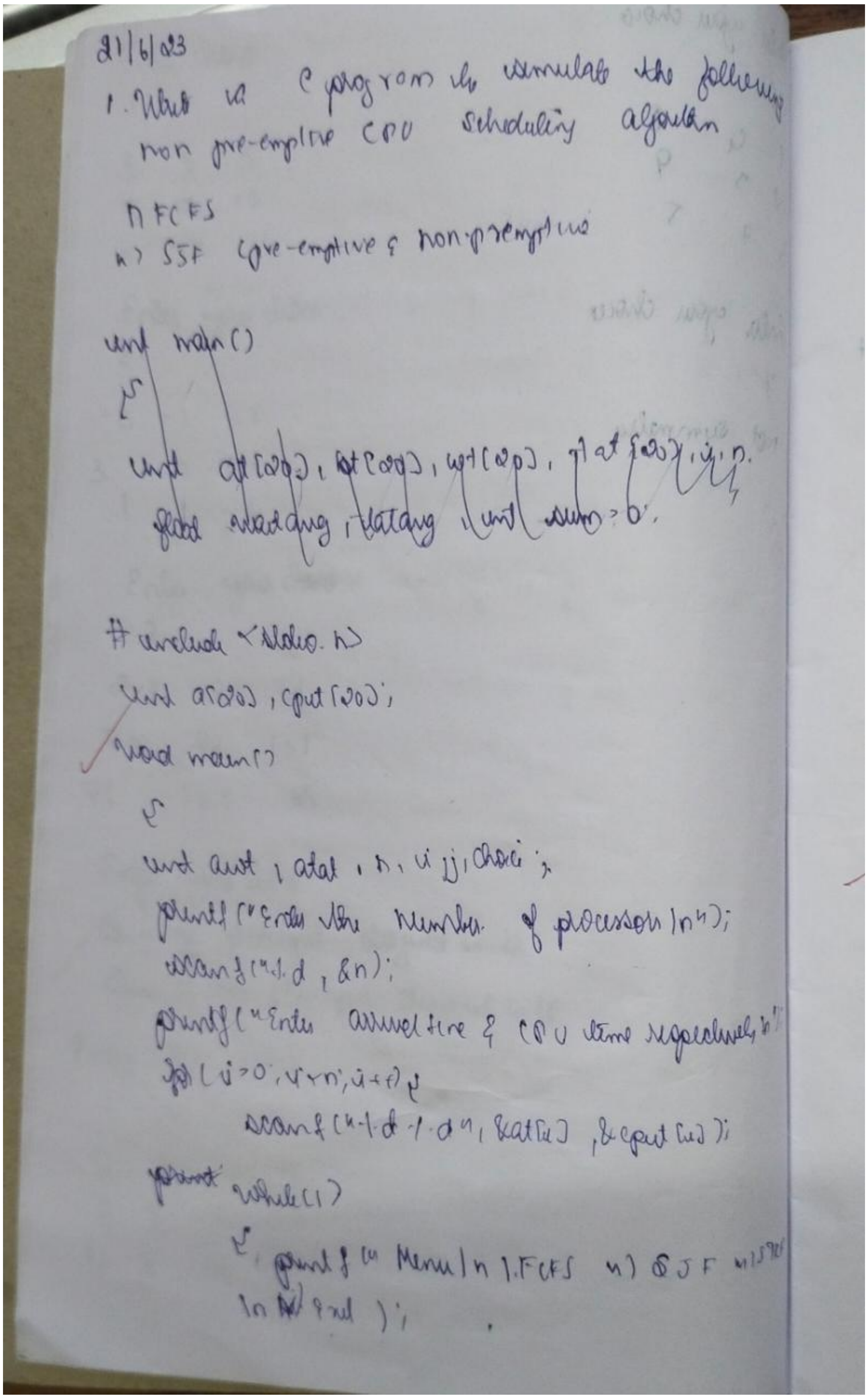
OS LAB-2

Q) Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time. FCFS SJF (pre-emptive & Non-pre-emptive

CODE-



21/6/23

1. Write a C program to simulate the following non pre-emptive CPU scheduling algorithm

i) FCFS
ii) SJF (pre-emptive & non-preemptive

```
~~int main()
{
int at[20], bt[20], wt[20], tat[20], i, j, n;
  float awtavg, tatavg, int sum=0;~~

#include <stdio.h>
int at[20], cput[20];
void main()
  {
  int awt, atat, n, i, j, choice;
   printf("Enter the number of process \n");
   scanf("%d", &n);
   printf("Enter arrival time & CPU time respectively\n");
   for (i=0; i<n; i++) {
          scanf("%d %d", &at[i], &cput[i]);
~~print~~ while(1)
       { printf("Menu\n 1.FCFS  2) SJF n15PR
       \n 4) Exit ");
~~scan~~ printf("\n Enter choice\n");
        scanf("%d", &choice);
    switch (choice ){
    case 1: fcfs(n);
            break;

    case 2: sjf(n);
            break;

    case 3: SRTf(n);
     break;

    }
  }

void fcfs (int n)

  int cpnt[20], tat[20], wt[20];
  float awt =0 atat = 0;

  int sum=0,i;
  for (i=0; i<n; i++){
       sum+= cput[i];
        cpnt [i]=sum;
       }
  for(i=0; i<n; i++){
         tat[i]= cpnt[i]- at[i];
         wt[i] = tat[i]- cput[i];

     }

  for(i=0; i<n;i+f){
                 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 Turn around time \n");

for(i=0; i<n; i++)
{
printf("\n \t P %d \t %d \t %d \t %d \t %d \t
      %d", i, at[i], cput[i], wt[i], tat[i]);

}
printf("\n\n Average Waiting Time --%f", awt)
printf("\n Average Turnaround Time --%f", atat);
}
void sjf(int n) {
    int cput[20], tat[20], wt[20];
     float awt=0, atat=0, sum_bt=0;
     int sum=0, i, j, smallest, temp=0;
    for(i=0; i<n; i++){
      sum_bt += cput[i];
    }
    cput[9] = 9999;
     while(sum < sum_bt).
    {
    smallest = 9
    for (i=0; i<n; i++){
       if (at[i]<= sum && cput[i] > 0 &&
        cput[i]< cput[smallest])
    }.      smallest = i;
```

```
printf("P[%d]\t\t%d\t%d\n", smallest, sum
   + cpu[smallest] - at[smallest], sum - at[smallest]);

  awt += sum + cpu[smallest] - at[smallest];
    atat += sum - at[smallest];
    sum += cpu[smallest];
     cpu[smallest] = 0;
  }

  awt = awt / n;
 atat = atat / n;
   printf("\n Average Waiting - %f", awt);
 printf("\n Average Turnaround time - %f", atat);
  }

~~void sortfcfs(){~~

void  srtf(int n)
   ~~int comp=0;i~~
   int remaining[n] = 0
     int curr = 0;
     int comp = 0;
     for(int i = 0; i<n; i++)
       { remaining[i] = cpu[i];
  }
     while (comp != n) {
               int shortest = -1
```

~~printf("Average waiting time %d", &awt)~~

~~printf("\n Average turnaround %d", &atat);~~

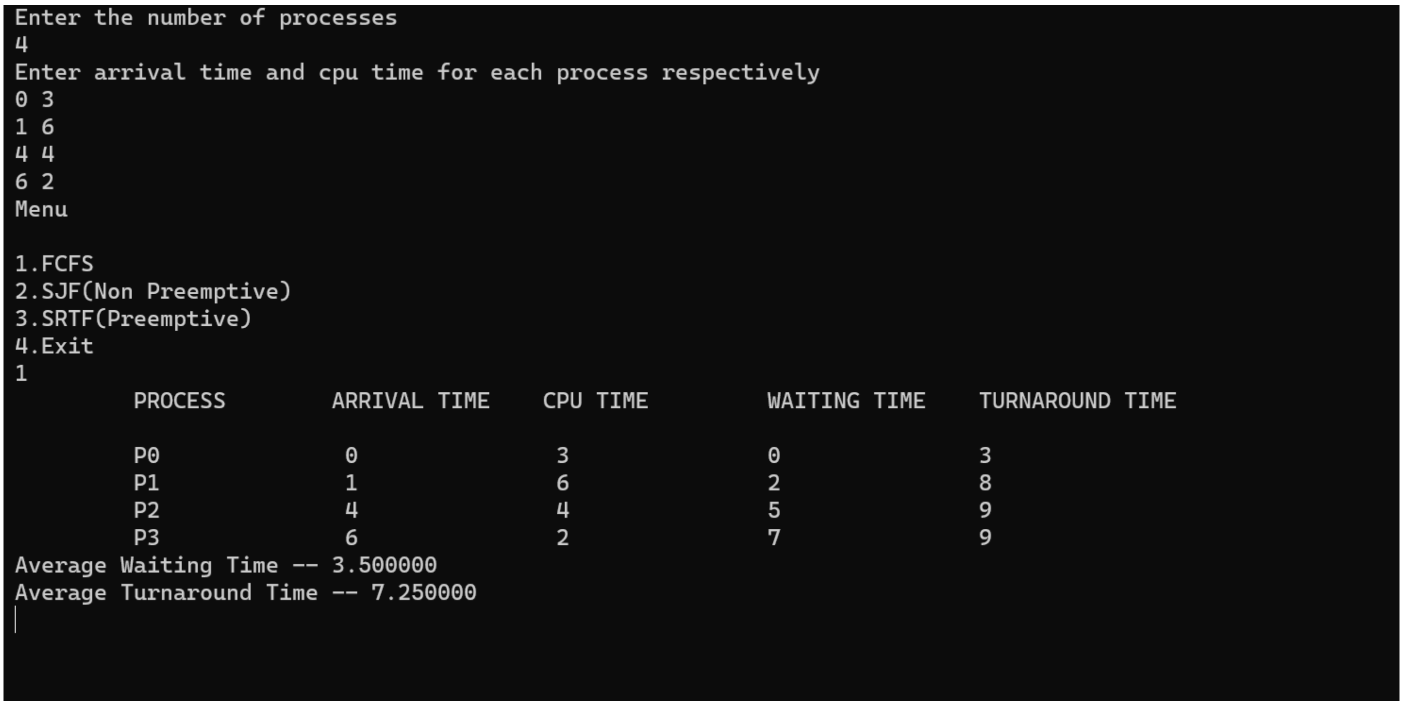
~~}~~

~~}~~

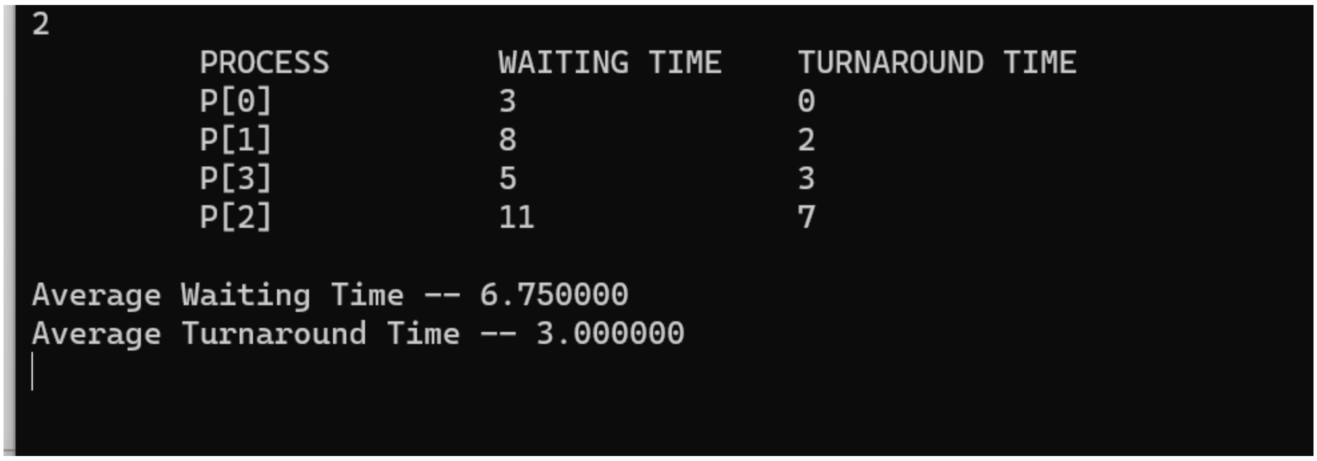
```
void srtf(int n)
  {
  int rt[20], tat[20], wt[20],
  c[20], smallest, time, i, count =0;
  float awt = 0, atat =0;
  for(i=0; i<n; i++)
    rt[i]= cpu[i];
  time =0;
  while(count != n) {
        smallest =-1;
        for(i =0; i<n; i++) {
      if (awt[i] <= time && rt[i] >0)
      if (smallest == -1 ||rt[i] < rt[smallest])
          smallest = i;
      }
    }
          if (smallest ==-1) {
                 time++;
                continue;   }
```

write legibly

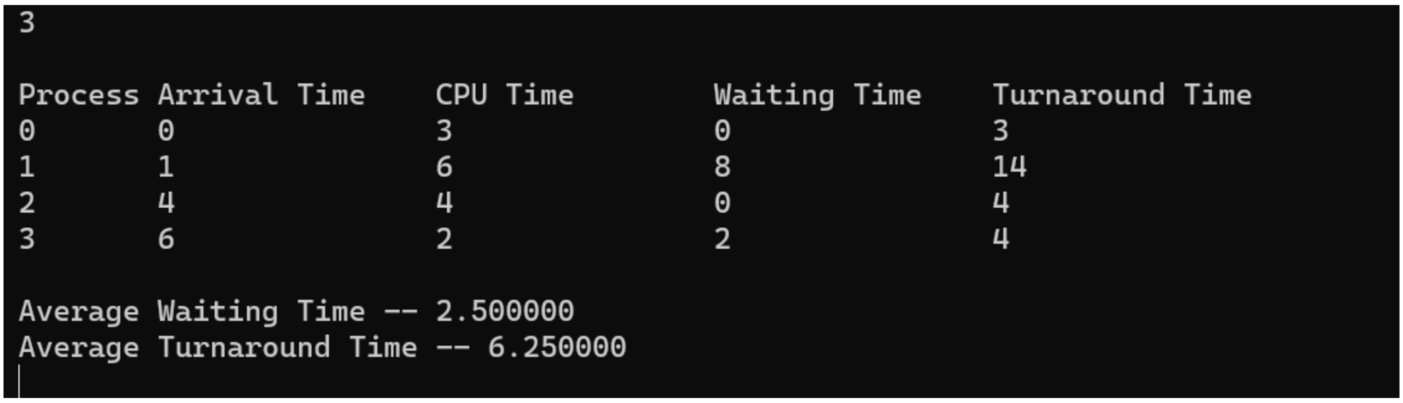
OUTPUT-



```
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

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
```