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:
#include <stdio.h>
int at[10], pt[10], ia, ip, n;
int tat[10], wt[10], it, iw, pos, j, i;
float atat = 0, awt = 0;
void fcfs()
{
  int t;
   printf("Enter number of processes: ");
   scanf("%d", &n);
  printf("Enter arrival times:\n");
  for (ia = 0; ia < n; ia++)
     scanf("%d", &at[ia]);
   printf("Enter process times:\n");
  for (ip = 0; ip < n; ip++)
     scanf("%d", &pt[ip]);
  if (at[0] == at[1])
     t = pt[1];
     pt[1] = pt[0];
     pt[0] = t;
  }
  if (at[0] != 0)
     tat[0] = at[0];
  for (it = 0; it < n; it++)
     tat[it] = 0;
  int i = 0;
  for (it = 0; it < n; it++)
     while (i <= it)
        tat[it] += pt[i++];
     i = 0;
  }
```

```
for (it = 0; it < n; it++)
     tat[it] = tat[it] - at[it];
  for (ia = 0; ia < n; ia++)
     wt[ia] = tat[ia] - pt[ia];
  for (i = 0; i < n; i++)
     atat += tat[i];
     awt += wt[i];
  }
  atat = atat / n;
   awt = awt / n;
  for (i = 0; i < n; i++)
  {
     printf("P%d\t%d\t%d\n", i, tat[i], wt[i]);
  }
  printf("Average TAT=%.2f\nAverage WT=%.2f\n", atat, awt);
}
void srtf()
  int rt[10], endTime, i, smallest;
  int remain = 0, time, sum_wait = 0, sum_turnaround = 0;
   printf("Enter no of Processes: ");
   scanf("%d", &n);
   printf("Enter arrival times\n");
  for (i = 0; i < n; i++)
  {
     scanf("%d", &at[i]);
   printf("Enter Process times \n");
  for (i = 0; i < n; i++)
     scanf("%d", &pt[i]);
     rt[i] = pt[i];
  rt[9] = 9999;
  for (time = 0; remain != n; time++)
     smallest = 9;
     for (i = 0; i < n; i++)
```

```
if (at[i] <= time && rt[i] < rt[smallest] && rt[i] > 0)
        {
          smallest = i;
     rt[smallest]--;
     if (rt[smallest] == 0)
        remain++;
        endTime = time + 1;
        printf("\nP%d %d %d", smallest + 1, endTime - at[smallest], endTime - pt[smallest] -
at[smallest]);
        sum wait += endTime - pt[smallest] - at[smallest];
        sum_turnaround += endTime - at[smallest];
     }
  }
  printf("\n\nAverage waiting time = %f\n", sum_wait * 1.0 / n);
  printf("Average Turnaround time = %f", sum_turnaround * 1.0 / n);
}
void sjf()
  int completed = 0;
  int currentTime = 0;
  int complete[n], ct[n];
  printf("Enter number of processes: ");
  scanf("%d", &n);
  printf("Enter arrival times:\n");
  for (int ia = 0; ia < n; ia++)
     scanf("%d", &at[ia]);
  printf("Enter process times:\n");
  for (int ip = 0; ip < n; ip++)
     scanf("%d", &pt[ip]);
  for (int i = 0; i < n; i++)
     complete[i] = 0;
     ct[i] = 0;
  }
  while (completed != n)
     int shortest = -1;
```

```
int min_bt = 9999;
   for (int i = 0; i < n; i++)
     if (at[i] <= currentTime && complete[i] == 0)</pre>
        if (pt[i] < min_bt)</pre>
        {
           min_bt = pt[i];
           shortest = i;
        }
        if (pt[i] == min_bt)
           if (at[i] < at[shortest])
              shortest = i;
        }
     }
  }
   if (shortest == -1)
     currentTime++;
  }
  else
   {
     ct[shortest] = currentTime + pt[shortest];
     tat[shortest] = ct[shortest] - at[shortest];
     wt[shortest] = tat[shortest] - pt[shortest];
     complete[shortest] = 1;
     completed++;
     currentTime = ct[shortest];
  }
}
for (int i = 0; i < n; i++)
   atat += tat[i];
   awt += wt[i];
}
atat = atat / n;
awt = awt / n;
for (int i = 0; i < n; i++)
```

```
{
     printf("P%d\t%d\t%d\n", i, tat[i], wt[i]);
  }
  printf("\nAverage TAT = %f\nAverage WT = %f\n", atat, awt);
}
void main()
  int op = 1, x;
  printf("1.FCFS \n2.SJF \n3.SRTF\n");
  scanf("%d", &x);
  switch (x)
  {
  case 1:
     fcfs();
     break;
  case 2:
     sjf();
     break;
  case 3:
     srtf();
     break;
  default:
     printf("Invalid option \n");
  }
}
```

## Output:

```
PS D:\VS Code\OS> cd "d:\VS Code\OS\"; if ($?) { gcc os.c -o os }; if ($?) { .\os }
1.FCFS
2.SJF
3.SRTF
1
Enter number of processes: 3
Enter arrival times:
0 0 1
Enter process times:
8 4 1
P0 4 0
P1 12 4
P2 12 11
Average TAT-9.33
Average WT-5.00
```

```
PS D:\US Code\OS> cd "d:\US Code\OS\" ; if ($?) { gcc os.c -o os } ; if ($?) { .\os }
1.FCFS
2.SJF
3.SRTF
2
Enter number of processes: 3
Enter arrival times:
0 0 1
Enter process times:
8 4 1
P0 13 5
P1 4 0
P2 4 3

Average TAT = 7.0000000
Average WI = 2.666667
PS D:\US Code\OS> 1
```

```
PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc os.c -o os } ; if ($?) { .\os }

1.FCFS
2.SJF
3.SRTF
3
Enter no of Processes : 3
Enter arrival times
0 0 1
Enter Process times
8 4 1

P3 1 0
P2 5 1
P1 13 5

Average waiting time = 2.000000
Average Turnaround time = 6.333333
PS D:\VS Code\OS = 1
```

(A) a 71-09-7002 c programing to simulate cpv achduling Por (1=0 algorin me 95 p 1 Pion fifse) (iii) Srjz comp ex me 1=0 while ( for ( It = 01 it chi it +++) tat[#]=0; tor (14=0; itch; 1++1) while (ic=it) ([++1]tq=+(+1) tat i=0', Por (i+ =0: iten: i+++) sidama 12 teitlit] = toutlit] - atlit]; for l'azo; lacu; iarr) pt lia Jigol sour for (120; iene; 147) 1 atatt = Lat (i ]; aux += wt(i); atat = atatm; aw+ = aw+In; for (1=0; ien; i++) i printer a pad you dod in", i, too (i), wt(i)); privit (" Average TAT=0), I in Average was 0/0 f v", and, 4. World Sif () } int completed =0; int currentline = 0: int complete (n), cf(n); printf (" futer number of processes" Sland (" olad ", don); point (" Enter provide times: "); Stort ( " of for (i=o; ien; itt) Scant (" o/vol ", fort (i)); prints (" Enter process times.

for ciro; icu; itt)

Sland (" o/ud", Lopt [i]); alai aw

```
Gold
                   complete (i)=0',
                   e+ (:)=0;
                while (completed !=n)
                     int glostes = -1;
                      int min_H = 9999;
                     for (1=0) (cn) (++)
                        it (at (i) <= current line 46 complete (i)==0)

d it (pt(i) <min-bt)
                                min_5+= pt (i);
                              4 Shortest =1; (1) 14 (6) 83
                            (+d-nino == (1)+1)
                             it (at (i) cat (shortest)
                                  d Storen=i; Pothests
);((
, fu", adat,
au+ );
                       Ct[Shortost] = currentline + pt[shortest];
fat[shortost] = ct[shortost] - at[shortost);
                       est [shortest] = tat[shortest] - pt[shortest];
                       Complete [shotest]=1;
                   completed + ;
                       Currentline = ct (86xtest);
               for (1=0, iens i++) = PAT and growth in
                 d atout += tat (i);
                    aust += wat (i);
               alat = atent /n;
               aux = aux /n;
                  Private (" Polodit old told in", ", tooti), wat(i));
              for (izo: Icn; itt)
```

```
print f (" in Average TAT = "lot in Average wit = "lot in", adaption
                                                                         L. FCFS
                                                                         2. SJF
                                                                          3. SRIF
                erteco
           Void
                                                                         1. Enter r
                 int at (10), evoltine, i, smallert;
                 int veneines, time, Sum-wait = 0, Sum-turnarousles
                                                                           Enter
                 printf("Enter number of processes: "):
                                                                           Enter
                Scarl (" God", Con);
                print I [" Enter arrival time in " 17 delle
                                                                            PI
                 for (1 =0; icn; i++)
                                                                            Pz
                   Scant (" " old ", Got(i));
                portut f (" Enter processes time in "7; (1) to ) ::
                                                                            P3
                    to (i=0; icn; 1++)
                                                                            AWA
                                                                             AWT
                      1 Scart (" 010d", 4 pt[i]);
                        2 8+(1) = pt(i);
                 of [9] = 9999; (+d along = - (1)+9) +
                  for (thre =0) renen :=n; time++)
                       gmallest=9;
                        for (i=o; icn; i+t)
                                                                              PI
                            if (at(i) <= time 46 of(i) < rf (smallest) db of(i) >0)
                                                                              122
                                                                              P3
                                 smellest = 1 1
                                                                              ATAT
                        of (smallest)-;
                                                                               TCI
                         it ( &+ (smallest = =0)
                           remain ++;
Re
                            end Time = time +1;
                        printf (unp fod god god , smallest, end Time-
                                                                             En
                        at [smallest, end time - pt [smallest] - at [smallest]);
                                                                              retuin
                     Sum-wait += end Time - pt [smallest] at [smallest];
                                                                               PI
                      Sum turnaroud + - evol Time - at (smallest);
                                                                                PZ
            project f(" in Average waiting time = 0/0 f (n", Sume turnarand
                                                                                P.3
                                                                               ATI
                                                                                AL
```

L. FCPS or C magnameto Simulado 282 sica 2. SJF 3, SPT F 1. Enter no. of processes 3. I have contract country Enter Arrival times 0 01 18019 18 19 Enter process times 8 ex 1 P1 12 4 PL 4 P3 12 11 A TPAT = 9.3. AW7 = 5. 2. Enter no. of processes 3. Enter Arrival time 0 0 1 Enter process times & PI 13 5 10<(i)+8 d 0 4 122 4 3 P3 F = TATA AWT = 2.67. 3. no. 9 processes 3. Enter Arrival tims 0 1 1Time-Enter 1 4 process times 8 ;((Ctr Entor woll; PI 5 13 0 PZ 4 3. 4 P3 ) (n) ATAT = 7 AW7 = 2067.