3.Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories ± system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

Code:

```
#include <stdio.h>
int spat[10], upat[10], i, n1, n2, p1[10], p2[10];
int sppt[10], uppt[10], time = 0, op = 0, y, z, pt;
int sptat[10], uptat[10];
int spwt[10], upwt[10];
float spatat = 0, spawt = 0;
float upatat = 0, upawt = 0;
void process(int x, int isSystem) {
  if (isSystem) {
     op += sppt[x];
     sptat[x] = op - spat[x];
     sppt[x] = 0;
     spwt[x] = sptat[x] - p1[x];
     spatat += sptat[x];
     spawt += spwt[x];
  } else {
     op += uppt[x];
     uptat[x] = op - upat[x];
     uppt[x] = 0;
     upwt[x] = uptat[x] - p2[x];
     upatat += uptat[x];
     upawt += upwt[x];
  }
}
int main() {
  printf("Enter the number of System Processes: ");
  scanf("%d", &n1);
  printf("Enter the number of User Processes: ");
  scanf("%d", &n2);
  printf("Enter the arrival times for System Processes:\n");
  for (i = 0; i < n1; i++)
```

```
scanf("%d", &spat[i]);
printf("Enter the process times for System Processes:\n");
for (i = 0; i < n1; i++)
  scanf("%d", &sppt[i]);
printf("Enter the arrival times for User Processes:\n");
for (i = 0; i < n2; i++)
  scanf("%d", &upat[i]);
printf("Enter the process times for User Processes:\n");
for (i = 0; i < n2; i++)
  scanf("%d", &uppt[i]);
for (i = 0; i < n1; i++)
  time += sppt[i];
for (i = 0; i < n2; i++)
  time += uppt[i];
for (i = 0; i < n1; i++)
  p1[i] = sppt[i];
for (i = 0; i < n2; i++)
  p2[i] = uppt[i];
printf("\n");
while (op < time) {
  y = -1;
  z = -1;
  for (i = 0; i < n1; i++) {
     if (op >= spat[i] \&\& sppt[i] != 0) {
        y = i;
        break;
     }
  for (i = 0; i < n2; i++) {
     if (op >= upat[i] && uppt[i] != 0) {
        z = i;
        break;
     }
  if (y != -1) {
     printf("%d SP%d ", op, y + 1);
```

```
process(y, 1);
     } else if (z != -1) {
        printf("%d UP%d ", op, z + 1);
        process(z, 0);
     } else {
        op++;
  }
  printf("%d ",op);
  printf("\n");
  printf("System Processes:\n");
  for (i = 0; i < n1; i++)
     printf("SP%d %d %d\n", i + 1, sptat[i],spwt[i]);
  printf("ATAT(System Processes): %.2f\n", spatat / n1);
  printf("AWT(System Processes): %.2f\n", spawt/n1);
  printf("\n");
  printf("User Processes:\n");
  for (i = 0; i < n2; i++)
     printf("UP%d %d %d\n", i + 1, uptat[i], upwt[i]);
  printf("ATAT(User Processes): %.2f\n", upatat / n2);
  printf("AWT(User Processes): %.2f\n", upawt / n2);
  return 0;
}
```

Output:

```
■ C:\Users\STUDENT\Desktop\Rev047\MLQ\bin\Debug\MLQ.exe
                                                                                                                                    X
Enter the number of System Processes: 3
Enter the number of User Processes: 1
Enter the arrival times for System Processes:
0 0 10
Enter the process times for System Processes:
Enter the arrival times for User Processes:
Enter the process times for User Processes:
0 SP1 4 SP2 7 UP1 15 SP3 20
System Processes:
SP2 7 4
ATAT(System Processes): 7.00
AWT(System Processes): 3.00
User Processes:
UP1 15 7
ATAT(User Processes): 15.00
AWT(User Processes): 7.00
Process returned 0 (0x0) execution time : 59.114 s
Press any key to continue.
```

Observation:

```
or Write a c program to simulate multi-level queue scheduly
                                                                     for Li=0;
for system process with higher priority & user prous Ja
                                                                      & bime
FeFs algorithm for the proces in each queue.
                                                                         PIL
thinclude. (State . h).
                                                                     for (1=0',
 int spat [10], upat [10], ", n1, 12, p1[10], P2[10];
                                                                      of time +
 "int sppt(io), uppt(io), time =0, op=0, 4, 2, 10+;
                                                                        P2 (1.
 int spront (101), upront (101), upront (101);
 floot spatat=0, spawt-o upatato, upawt-o;
                                                                       private (
                                                                     while COP
 void process ( int or , int is system) of
      of (& System) of
                                                                            7
           op+= sppt [n];
                                                                          for
           sphot[n]= op-sport(n);
           sput Crij= Sprad Crij - pi[rij;
          Spawt += Spwt[n];
        elsed
             Op+=uppt[n];
            uptat [n] = op -upat(n];
             uppt[n]=0;
             "( ( m) cd - [ m) potat [ m ] - [m)
             upated += uplatery;
         4 upanot += upnot [a];
                                                                        1214
  in mainer }
      printf("Enter the number of System processes: ");
      Scanflyold", an1);
                                                                         else
     privat (" Enter the number of User processes: ");
     Sant (" olod", 602);
     printf("Enter arrival times for System Processes: 1n");
forci=0; i<n1; i+1)

grant('4001', GSpat(i));
                                                                        elsen
     printf(" Enter process times for system processes: "
                                                                   Poluto ( " olo
     for (10) ich jitt)
                                                                      mintel
      Seant ("'olod", & SpAt[i]);
                                                                       for li=
     printf(" Enter arrived times for User process; in"))
                                                                    Bring to MATA
       for (1=0; icn2; itt)
       scanf("11 od", & upat[i]).
                                                                    printe (" AW
                                                                     Driv4+ [H
    printf ("Enter prouss times for user prous: m");
scanf (" "lock", & upport (i));
```

```
( "M: comising only " 1 2 min)
             for 11=0; 12 mi, 1+4)
              d time += sppt(1);
                  PITIJ = SPP+(i]:
                                Cont Lobogu, "of to all : (70) 1414 " Laring
             for (120; 12n2; itt) coul tenoque, "as to of 1 (que south ) Mining
              of time == upot (i);
               PZ[1] = cappt[i]:
              print+("un");
             while (opetime) }
                    year processed in 1 - 24
                    7 =-1;
                  for ( i=0; ien ; i++) }
                         1+ (op >= Spat [i] 44 Sppt[i] != 0)
                                break ; roll at evit lovino
                    for (1=0; icn2; i++) {
                  14 (op >= uport [i] 46 uppt [i] !=0)
                            d /2=1)42 21 190 $ 492 # 192 0
                                break; 4
                12 (41=-1) 4
                     print f( " " od SP " od ", OP, 4+1);
                      process(y, 1);
                                                   - 00.8: (92) FATA
                 else 1+ (2; =-1) 1
                      Print("4.d Up 9.d ", op, zx1);
                      prous (2,0);
7;
                 else 1 Up++; 4
            project + (" olo d > ", op);
              Print+("System Processes: "\");

for (i=0; fcn); i++)

print+("Sp god wod wod m", (i+1), sptat(i), spw(i))
            print("ATATCSP): "10.28 in", spatat/ni);
print("Awt(sp); 10.24 w", spawt/ni);
              のかれましれいかりつ
```

printer User processes: (M"); print [" upoled "Ind red in", it, uptact [i], uput [i]; forti=0; icn2; 1++) private" ATAT (UP): 010,24 m, upatout /m); printf("ANT (UP): 10.24 in", upawt In2); return 0; 4. output : Enter number of System processes: 3 Enter number of user processes: 1 Enter arrival times for System Processes: 00 10 Enter process times for System processes: Enter arrival time for Wer processes: Enter process times for user processes 0 SP1 4 SP2 7 UP1 15 SP3 20 System processes. SPI 10 5 SP3 ATAT (SP) : 7.00 . AWT (SP): 3.00. User porocuses: UP1 15 7 00-71: (4U) TATA 4WT (UP): 7.00.