

Write a C program to simulate multi-level queue scheduling algorithm cons. scenario → All processes are divided into SP and UP. SP to be given more priority than UP. Use facts.

```
#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, x, pt;
int spatat[10], upat[10];
int spwt[10], upwt[10];
float spatat=0, spawt=0;
float upat[10], upawt=0;
void process(int x, int isSystem) {
    if (isSystem) {
        op += sppt[x];
        spat[x] = op - spat[x];
        sppt[x] = 0;
        spwt[x] = spat[x] - p1[x];
        spatat += spat[x];
        spawt += spwt[x];
    } else {
        op += uppt[x];
        upat[x] = op - upat[x];
        uppt[x] = 0;
        upwt[x] = upat[x] - p2[x];
        upat[10] = upat[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:");
    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] = spat[i];
for(i=0; i<n2; i++)
p2[i] = uppt[i];
printf("\n");
while (OP < time) {
    y = -1;
    x = -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) {
            x = i;
            break;
        }
    }
}

```



```

if (y != -1)
{
    printf (" %d SP %d ", op, y+1);
    Process (y, 1);
}
else if (x != -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, spatat[i], spwt[i]);
printf ("ATAT (system Processes): %2f \n", spatat/n1);
printf ("\n");
printf ("User Processes: \n");
for (i = 0; i < n2; i++)
    printf ("UP %d %d %d \n", i+1, upat[i], upwt[i]);
printf ("ATAT (user Processes): %2f \n", upat/n2);
printf ("AWT (user Processes): %2f \n", upawt/n2);
return 0;
}

```

Output

Enter the number of system Processes : 3

Enter the number of user Processes : 2

Enter the arrival times for System Processes:

1

4

2

Enter the process times for System Processes :

5

8

9

Enter the arrival times for user Processes:

0

5

Enter the process times for user processes

6

4

0 UP1 6 SP1 11 SP2 19 SP3 28 UP2 32

System Processes:

SP1 10 5

SP2 15 7

SP3 26 17

ATAT (system Processes): 17.00

AWT (system Processes): 9.67

User Processes:

UP1 6 0

UP2 27 23

ATAT (user Processes): 16.50

AWT (user Processes): 11.50.

8/10

19/7/23

```
Enter the number of System Processes: 3
Enter the number of User Processes: 2
Enter the arrival times for System Processes:
1
4
2
Enter the process times for System Processes:
5
8
9
Enter the arrival times for User Processes:
0
5
Enter the process times for User Processes:
6
4

0 UP1 6 SP1 11 SP2 19 SP3 28 UP2 32
System Processes:
SP1 10 5
SP2 15 7
SP3 26 17
ATAT(System Processes): 17.00
AWT(System Processes): 9.67

User Processes:
UP1 6 0
UP2 27 23
ATAT(User Processes): 16.50
AWT(User Processes): 11.50
```