2)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.

```
#include <stdio.h>

int main() {
   int n, i, type[20], bt[20], pid[20];
   int sys_bt[20], sys_pid[20], user_bt[20], user_pid[20];
   int sys_n = 0, user_n = 0;
   int wt[20], tat[20];
   int time = 0;

   printf("Enter total number of processes: ");
   scanf("%d", &n);

   for(i = 0; i < n; i++) {
      printf("Enter Process ID, Burst Time, Type (0=System, 1=User) for P[%d]: ", i+1);
      scanf("%d%d%d", &pid[i], &bt[i], &type[i]);
      if(type[i] == 0) {
         sys_pid[sys_n] = pid[i];
         sys_bt[sys_n] = bt[i];
         sys_n++;
      } else {
         user_pid[user_n] = pid[i];
         user_bt[user_n] = bt[i];
         user_n++;
      }
   }

   printf("\nExecuting System Processes (High Priority - FCFS):\n");
   for(i = 0; i < sys_n; i++) {
      wt[i] = time;
      tat[i] = wt[i] + sys_bt[i];
      time += sys_bt[i];
      printf("P[%d] (System) - BT: %d, WT: %d, TAT: %d\n", sys_pid[i], sys_bt[i], wt[i], tat[i]);
   }

   printf("\nExecuting User Processes (Low Priority - FCFS):\n");
   for(i = 0; i < user_n; i++) {
```

```
        wt[sys_n + i] = time;
        tat[sys_n + i] = wt[sys_n + i] + user_bt[i];
        time += user_bt[i];
        printf("P[%d] (User) - BT: %d, WT: %d, TAT: %d\n", user_pid[i], user_bt[i], wt[sys_n + i],
tat[sys_n + i]);
    }

    float avg_wt = 0, avg_tat = 0;
    for(i = 0; i < sys_n + user_n; i++) {
        avg_wt += wt[i];
        avg_tat += tat[i];
    }

    printf("\nAverage Waiting Time = %.2f", avg_wt / (sys_n + user_n));
    printf("\nAverage Turnaround Time = %.2f\n", avg_tat / (sys_n + user_n));

    return 0;
}
```

Output Clear

```
Enter total number of processes: 2
Enter Process ID, Burst Time, Type (0=System, 1=User) for P[1]: 5
6
9
Enter Process ID, Burst Time, Type (0=System, 1=User) for P[2]: 6
3
8

Executing System Processes (High Priority - FCFS):

Executing User Processes (Low Priority - FCFS):
P[5] (User) - BT: 6, WT: 0, TAT: 6
P[6] (User) - BT: 3, WT: 6, TAT: 9

Average Waiting Time = 3.00
Average Turnaround Time = 7.50


=== Code Execution Successful ===
```