CS 3413

Assignment 1

Due Date: September 11th, 2019 at 9:30 am

ASSIGNMENT IS TO BE COMPLETED INDIVIDUALLY BY ALL STUDENTS!

In the early days of computers, operating systems serviced users by scheduling jobs in the order in which they arrived (started). This is a First-Come First-Serve (FCFS) algorithm.

The downside of First-Come First-Serve (FCFS) is that you are at the mercy of the order in which jobs arrive. A large job at the beginning has everyone waiting and becoming irritated. A solution to this is to try and help people with small jobs get in and out quickly. This is called Shortest-Job-First (SJF). In this algorithm as jobs arrive they are added to the run list based on the duration of the job – shortest duration first! When the duration of multiple jobs are the same then the arrival time decides who goes first.

To properly implement SJF you also need pre-emption – the concept of periodically stopping a job to take away the CPU and decide who is the next eligible process to run. This is needed to prevent large jobs that acquire the CPU and start computing to only hold the CPU while new short jobs arrive and have to wait. To implement pre-emption, you only execute a job for 1 period and then you add it back into the run list.

For example if we have 1 CPU:

User	Process	Arrival	Duration
Jim	A	2	5
Mary	B	2	3
Sue	D	5	5
Mary	C	6	2

This would result in:

Time	Job
2	B
3	B
4	B
5	A
6	C
7	C
8	A
9	A
10	A
11	A
12	D
13	D
14	D

```
15          D
16          D
17          IDLE

Summary
Jim         12
Mary        8
Sue         17
```

Write a C program that will read in job requests and print out the corresponding job schedule according to a SJF algorithm as above. The input format is each line entry contains a job separated by a tab. The first line of input is ignored as the header. The output format is two tables. First table is the running time and the job currently executing (tab separated). The second table is a summary with the user name (in the order in which jobs arrive) and the time when their **last** job is completed.

NOTE: THE INPUT AND OUTPUT OF YOUR PROGRAM IS SPECIFIED SUCH THAT THE MARKER CAN AUTO TEST YOUR SUBMISSIONS. PLEASE FOLLOW THE SPECIFICATIONS! INPUT WILL BE READ VIA `stdin` (EG. KEYBOARD). OUTPUT SHOULD BE PRINTED TO `stdout` (EG. MONITOR).

YOU CAN FIND THE SAMPLE INPUT AND OUTPUT FILES WITH THE ASSIGNMENT ON D2L. BE SURE YOUR PROGRAM READS/PRINTS THE PROPER INPUT/OUTPUT FORMATTING!