**CSC 457**

# Assignment 5 Due: 11/6/2024 11:59 pm

**For this assignment, you will write a program in C. Make sure that you document your program (including comments that explain your program), use proper style (indentation!) and use meaningful names for variables and functions.**

**Teams: I encourage you to work in pairs to complete this assignment. Students working with a partner should each do their share. If you cannot find a partner, you can work alone.**

**If you choose to work in pairs, only one student needs to submit the assignment, and don't forget to include both of your names in the code and in the document containing the program output.**

---

**Program Description:**

Download the zip folder provided on Canvas and unzip it to get started. You are writing a (console) program to create several queues of (simulated, fake, simplified) process control blocks. **Linked lists must be used instead of arrays to accomplish this assignment.** The program should read from a file (***processes.txt***), one line for each process control block (format given below), determine

- whether it would have arrived at the OS (based on arrival time – 0 it has arrived, or later? – it has not arrived),
- and if arrived, whether it is real time, and what its priority is.

Add the process control block to the (**end of the) appropriate queue** – standby (if not arrived yet), real time, or priority 1, 2, or 3. After all input is read in,

- show all queues of processes,
- **"start" the first process** (see below) and then
- present the user options for seeing info, as discussed below.

**Background Information:**

- In this particular simulation, processes can be "real time" which requires very time-sensitive (if not immediate) action, and other priorities numbered 1,2, …. (Low numbers are high priority). The data is for all possible priorities.
- In all there will be 6 queues – standby, real time, priority 1, priority 2, priority 3, and exit queue.
- We are still working on implementing simulated CPU scheduling.
- **See sample interface interaction for an example of the program in action.**

**Task Details:**

- Each line of the file has the following (separated by a space) (A process file is provided. It contains 40 process information)
  - int processId;
  - int userId;
  - int arrivalTime; `// time units in simulation, not actual time`
  - int priority; `// base priority`
  - int expectedTimeRemaining;
  - int expectedMemoryNeed;
  - int expectedPctCPU; `// to get an idea of whether CPU bound or IO bound`
  - bool realTime; `// 0 or 1 - whether a real-time process or not (real-time processes may need immediate attention)`
- A `struct` called processblock is provided to hold process information plus the following two members.
  - State processState; `// Not input - initially NEW`
  - struct processblock *nextPtr; `// pointer to next process control block`
- **Starting the first process involves picking the process to run (as with replacing below) and pointing to the running process.**
- Users should be given options (see sample interaction) to allow them to choose including (each should also report how many processes match the criteria (see sample interaction)):
  - Running process finishes (it goes to the end of the **"Exit" queue**, and **is replaced**)
  - Running process's time is up for now (**it is replaced** and goes back in (the back of) its appropriate Ready queue).
  - See all processes for a particular specified priority (0-3, zero is real time) (This should be easy! – display the appropriate queue)
  - See Standby arrivals for time T (looking ahead to what is coming). NOTE: **equal to T, not less than or greater than**
  - See all processes with expected time remaining greater than a particular specified amount (on any of the Ready queues)
  - See all queues (except standby) – real time, priorities 1,2, and 3, and Exit queue
  - Quit

- **Replacing the running process (for options #1 and #2, plus picking the initial running process):**
  - Take the first process from the front of the real time queue (if there is one)
  - Else take the first process from the front of the priority 1 queue (if there is one)
  - Else take the first process from the front of the priority 2 queue (if there is one)
  - Else take the first process from the front of the priority 3 queue (if there isn't one, in this program you're done)
  - HINT: Make sure that you not only take the node off the linked list but also make sure it doesn't continue to point to its former next node
- Put a getchar(); at the end of your main program to pause to see results, and at other places where you might want to see what has happened.
- You may use the keyword `typedef` to create shorter names for `struct` types and simplify declarations of types like pointers. E.g.:
  - typedef struct processblock PCB; `// synonym for struct processblock`
  - typedef PCB* ptrPCB; `// synonym for PCB*`
- You should use functions to divide up the task. The following functions are required to create (the function/parameters are more important than the name):
  - read a line from the file and return a pointer to a newly allocated process control block. E.g. `ptrPCB readBlock(FILE *fPtr)`
  - add a node to the end of a list – passed a **pointer to the start of list pointer** and a pointer to the node to be added e.g.

    `void AddToEnd(ptrPCB *headRef, ptrPCB toAddPtr)`
  - delete a node from the front of a list of processblocks - passed - the pointer to the pointer to the start of the list

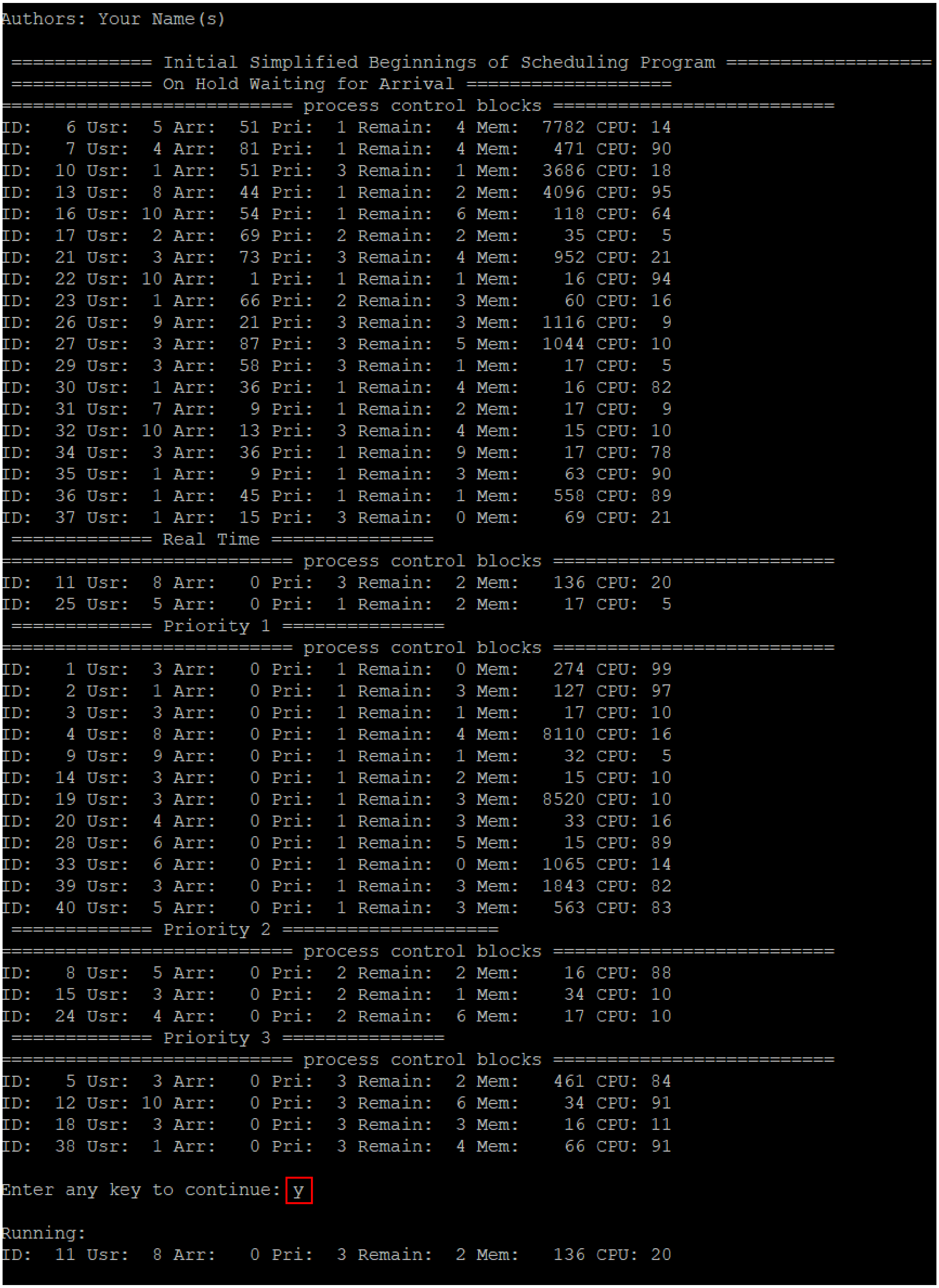
    `// basically take first element of list off and return it` e.g.

    `ptrPCB GetAndRemoveFront(ptrPCB *headRef)`
  - **The add and delete functions should be suitable for use with ANY of the 6 queues that you use** (with the Exit queue you'll only use add)
- You can also create other helper functions:
  - display one process control block
  - display all of the process control blocks in one queue (calling the above probably)
  - display the choices to the user and get a valid choice from them
- Make your program work as shown in the sample output. Display your name(s) during the user interaction.
- MAKE SURE YOUR PROGRAM WORKS! (i.e. gets the correct results) - it doesn't have to just run, it needs to run correctly!
- Put YOUR NAME(S), e-mail address(es), date, and purpose of the program in comments at the top of the code. The purpose should be what the program is supposed to do. You MUST include comments that explain your program to get full credit.

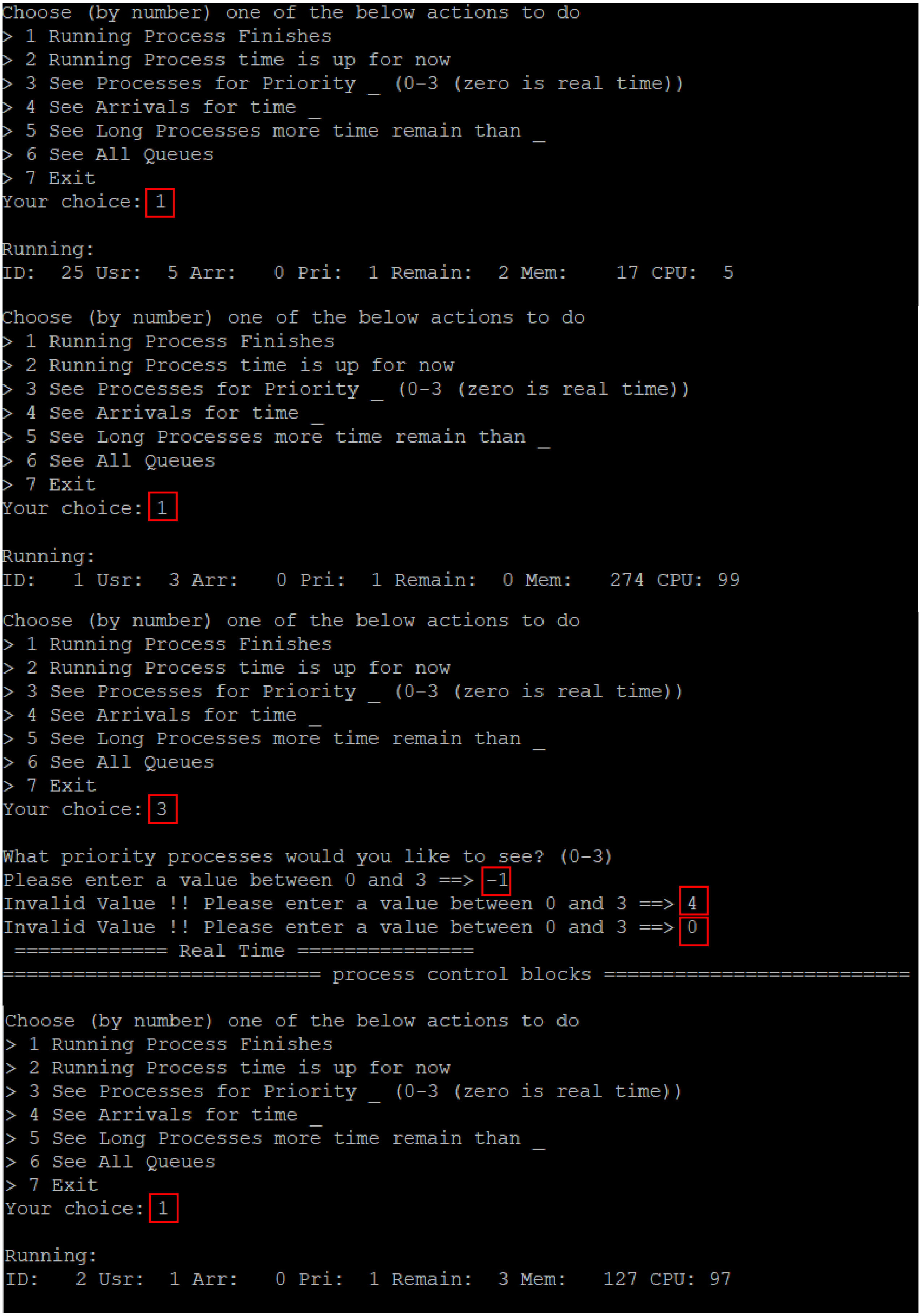
---

**The output of the sample interface interactions is on the following pages.**

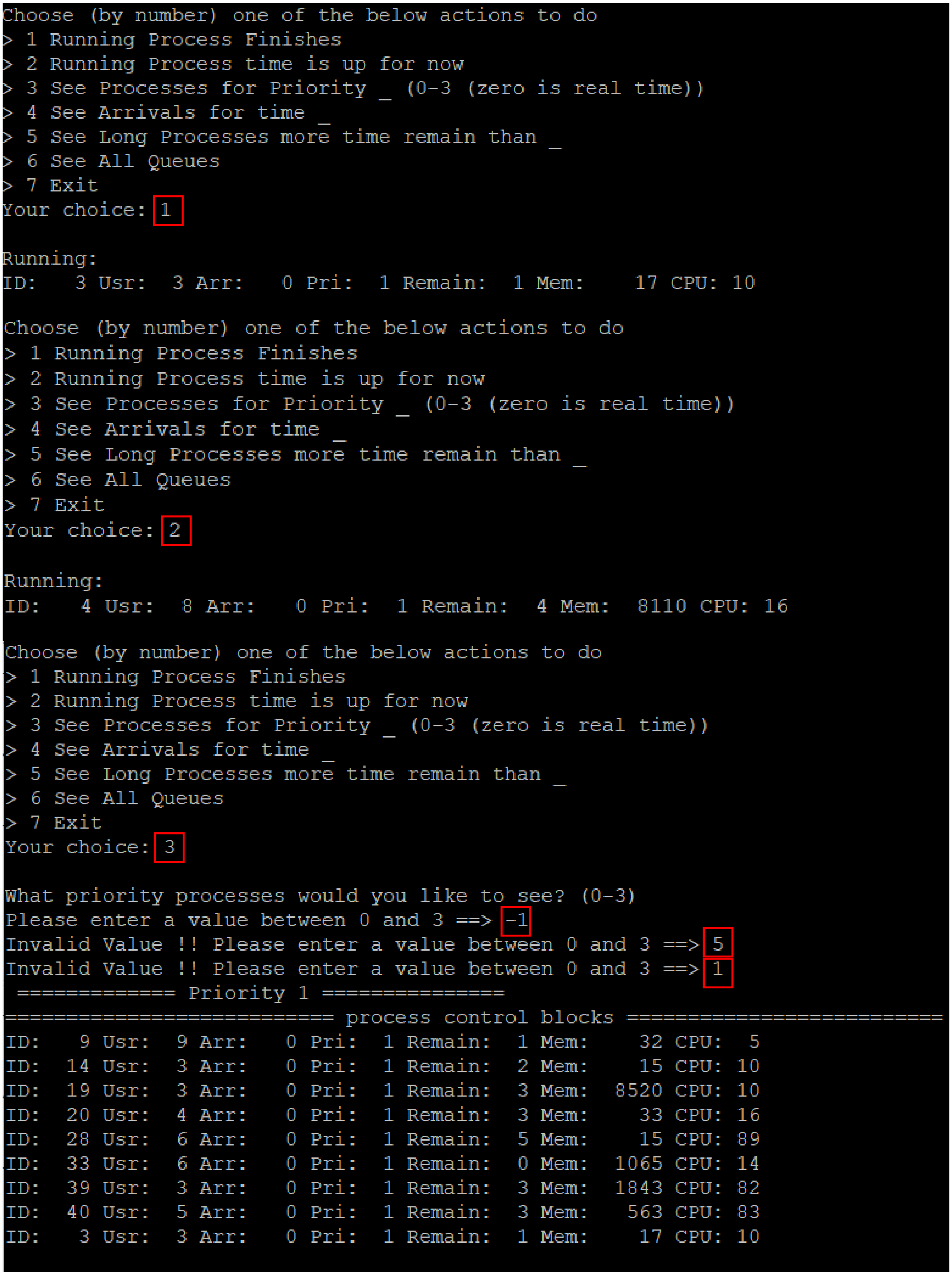
**Sample output: (The output here relies heavily on abbreviations. User input is highlighted in a red box.)**



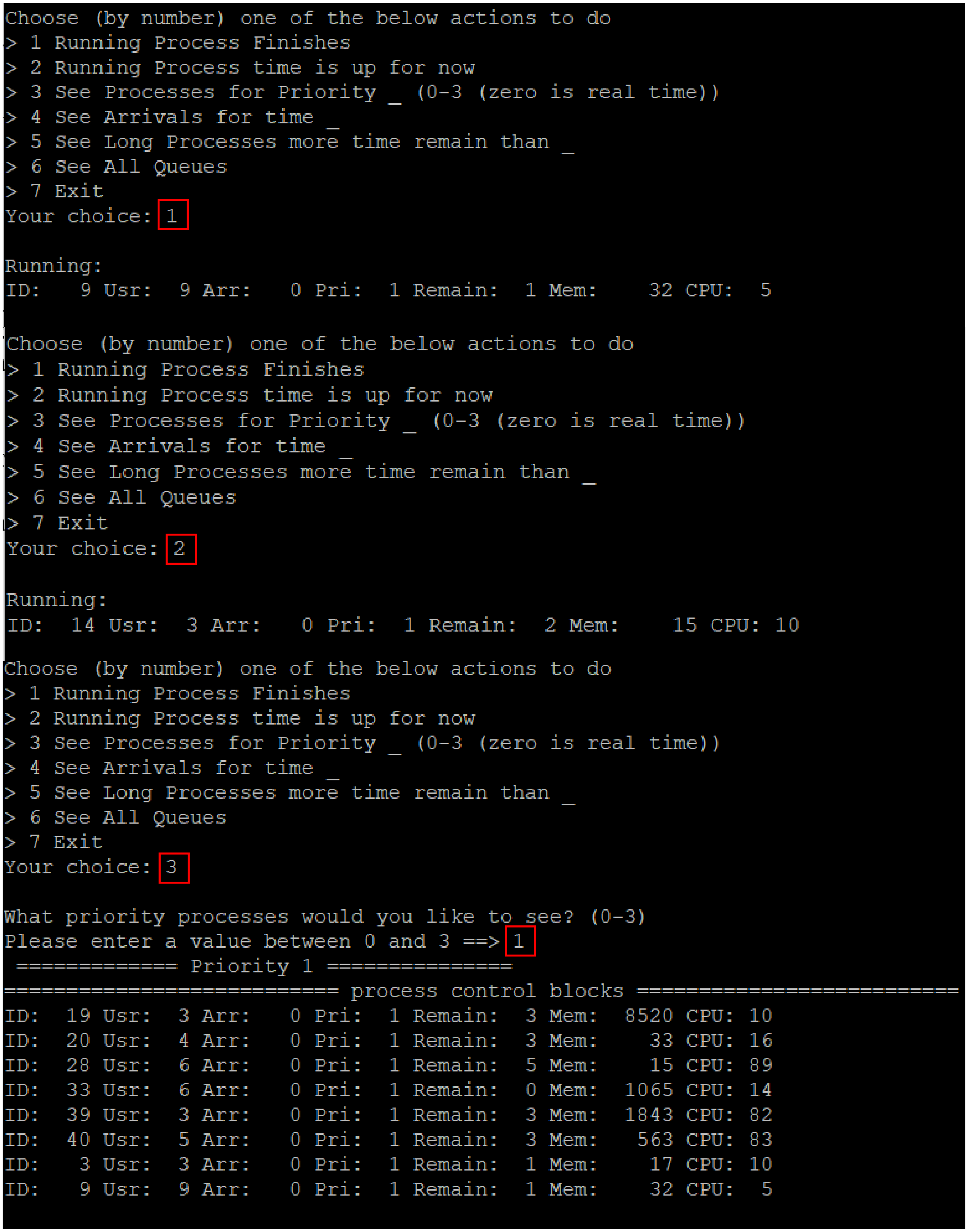
```
Authors: Your Name(s)

 ============= Initial Simplified Beginnings of Scheduling Program ===================
 ============= On Hold Waiting for Arrival ===================
=========================== process control blocks ==========================
ID:   6 Usr:  5 Arr:  51 Pri:  1 Remain:  4 Mem:  7782 CPU: 14
ID:   7 Usr:  4 Arr:  81 Pri:  1 Remain:  4 Mem:   471 CPU: 90
ID:  10 Usr:  1 Arr:  51 Pri:  3 Remain:  1 Mem:  3686 CPU: 18
ID:  13 Usr:  8 Arr:  44 Pri:  1 Remain:  2 Mem:  4096 CPU: 95
ID:  16 Usr: 10 Arr:  54 Pri:  1 Remain:  6 Mem:   118 CPU: 64
ID:  17 Usr:  2 Arr:  69 Pri:  2 Remain:  2 Mem:    35 CPU:  5
ID:  21 Usr:  3 Arr:  73 Pri:  3 Remain:  4 Mem:   952 CPU: 21
ID:  22 Usr: 10 Arr:   1 Pri:  1 Remain:  1 Mem:    16 CPU: 94
ID:  23 Usr:  1 Arr:  66 Pri:  2 Remain:  3 Mem:    60 CPU: 16
ID:  26 Usr:  9 Arr:  21 Pri:  3 Remain:  3 Mem:  1116 CPU:  9
ID:  27 Usr:  3 Arr:  87 Pri:  3 Remain:  5 Mem:  1044 CPU: 10
ID:  29 Usr:  3 Arr:  58 Pri:  3 Remain:  1 Mem:    17 CPU:  5
ID:  30 Usr:  1 Arr:  36 Pri:  1 Remain:  4 Mem:    16 CPU: 82
ID:  31 Usr:  7 Arr:   9 Pri:  1 Remain:  2 Mem:    17 CPU:  9
ID:  32 Usr: 10 Arr:  13 Pri:  3 Remain:  4 Mem:    15 CPU: 10
ID:  34 Usr:  3 Arr:  36 Pri:  1 Remain:  9 Mem:    17 CPU: 78
ID:  35 Usr:  1 Arr:   9 Pri:  1 Remain:  3 Mem:    63 CPU: 90
ID:  36 Usr:  1 Arr:  45 Pri:  1 Remain:  1 Mem:   558 CPU: 89
ID:  37 Usr:  1 Arr:  15 Pri:  3 Remain:  0 Mem:    69 CPU: 21
 ============= Real Time ===============
=========================== process control blocks ==========================
ID:  11 Usr:  8 Arr:   0 Pri:  3 Remain:  2 Mem:   136 CPU: 20
ID:  25 Usr:  5 Arr:   0 Pri:  1 Remain:  2 Mem:    17 CPU:  5
 ============= Priority 1 ===============
=========================== process control blocks ==========================
ID:   1 Usr:  3 Arr:   0 Pri:  1 Remain:  0 Mem:   274 CPU: 99
ID:   2 Usr:  1 Arr:   0 Pri:  1 Remain:  3 Mem:   127 CPU: 97
ID:   3 Usr:  3 Arr:   0 Pri:  1 Remain:  1 Mem:    17 CPU: 10
ID:   4 Usr:  8 Arr:   0 Pri:  1 Remain:  4 Mem:  8110 CPU: 16
ID:   9 Usr:  9 Arr:   0 Pri:  1 Remain:  1 Mem:    32 CPU:  5
ID:  14 Usr:  3 Arr:   0 Pri:  1 Remain:  2 Mem:    15 CPU: 10
ID:  19 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  8520 CPU: 10
ID:  20 Usr:  4 Arr:   0 Pri:  1 Remain:  3 Mem:    33 CPU: 16
ID:  28 Usr:  6 Arr:   0 Pri:  1 Remain:  5 Mem:    15 CPU: 89
ID:  33 Usr:  6 Arr:   0 Pri:  1 Remain:  0 Mem:  1065 CPU: 14
ID:  39 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  1843 CPU: 82
ID:  40 Usr:  5 Arr:   0 Pri:  1 Remain:  3 Mem:   563 CPU: 83
 ============= Priority 2 ====================
=========================== process control blocks ==========================
ID:   8 Usr:  5 Arr:   0 Pri:  2 Remain:  2 Mem:    16 CPU: 88
ID:  15 Usr:  3 Arr:   0 Pri:  2 Remain:  1 Mem:    34 CPU: 10
ID:  24 Usr:  4 Arr:   0 Pri:  2 Remain:  6 Mem:    17 CPU: 10
 ============= Priority 3 ===============
=========================== process control blocks ==========================
ID:   5 Usr:  3 Arr:   0 Pri:  3 Remain:  2 Mem:   461 CPU: 84
ID:  12 Usr: 10 Arr:   0 Pri:  3 Remain:  6 Mem:    34 CPU: 91
ID:  18 Usr:  3 Arr:   0 Pri:  3 Remain:  3 Mem:    16 CPU: 11
ID:  38 Usr:  1 Arr:   0 Pri:  3 Remain:  4 Mem:    66 CPU: 91

Enter any key to continue: y

Running:
ID:  11 Usr:  8 Arr:   0 Pri:  3 Remain:  2 Mem:   136 CPU: 20
```



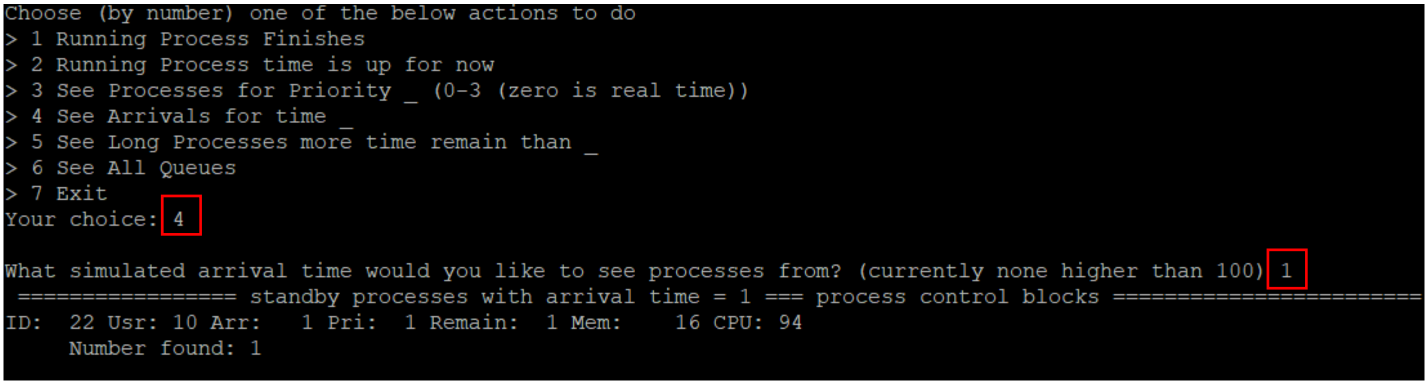
```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 1

Running:
ID:   25 Usr:   5 Arr:    0 Pri:   1 Remain:   2 Mem:     17 CPU:   5

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 1

Running:
ID:    1 Usr:   3 Arr:    0 Pri:   1 Remain:   0 Mem:    274 CPU:  99

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 3

What priority processes would you like to see? (0-3)
Please enter a value between 0 and 3 ==> -1
Invalid Value !! Please enter a value between 0 and 3 ==> 4
Invalid Value !! Please enter a value between 0 and 3 ==> 0
 ============= Real Time ===============
=========================== process control blocks ==========================

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 1

Running:
ID:    2 Usr:   1 Arr:    0 Pri:   1 Remain:   3 Mem:    127 CPU:  97
```



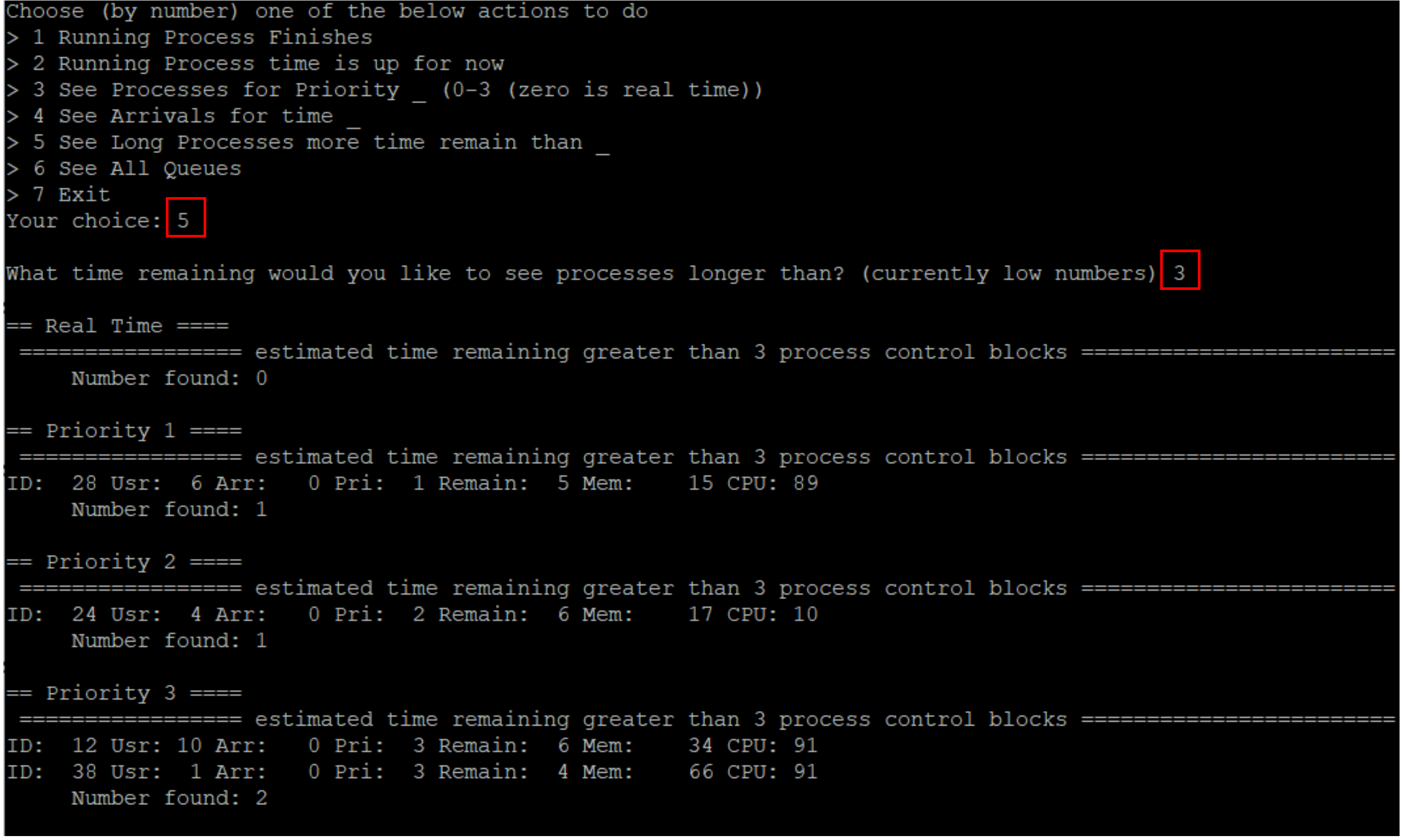
```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 1

Running:
ID:   3 Usr:  3 Arr:   0 Pri:  1 Remain:  1 Mem:    17 CPU: 10

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 2

Running:
ID:   4 Usr:  8 Arr:   0 Pri:  1 Remain:  4 Mem:  8110 CPU: 16

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 3

What priority processes would you like to see? (0-3)
Please enter a value between 0 and 3 ==> -1
Invalid Value !! Please enter a value between 0 and 3 ==> 5
Invalid Value !! Please enter a value between 0 and 3 ==> 1
 ============= Priority 1 ===============
=========================== process control blocks ==========================
ID:   9 Usr:  9 Arr:   0 Pri:  1 Remain:  1 Mem:    32 CPU:  5
ID:  14 Usr:  3 Arr:   0 Pri:  1 Remain:  2 Mem:    15 CPU: 10
ID:  19 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  8520 CPU: 10
ID:  20 Usr:  4 Arr:   0 Pri:  1 Remain:  3 Mem:    33 CPU: 16
ID:  28 Usr:  6 Arr:   0 Pri:  1 Remain:  5 Mem:    15 CPU: 89
ID:  33 Usr:  6 Arr:   0 Pri:  1 Remain:  0 Mem:  1065 CPU: 14
ID:  39 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  1843 CPU: 82
ID:  40 Usr:  5 Arr:   0 Pri:  1 Remain:  3 Mem:   563 CPU: 83
ID:   3 Usr:  3 Arr:   0 Pri:  1 Remain:  1 Mem:    17 CPU: 10
```



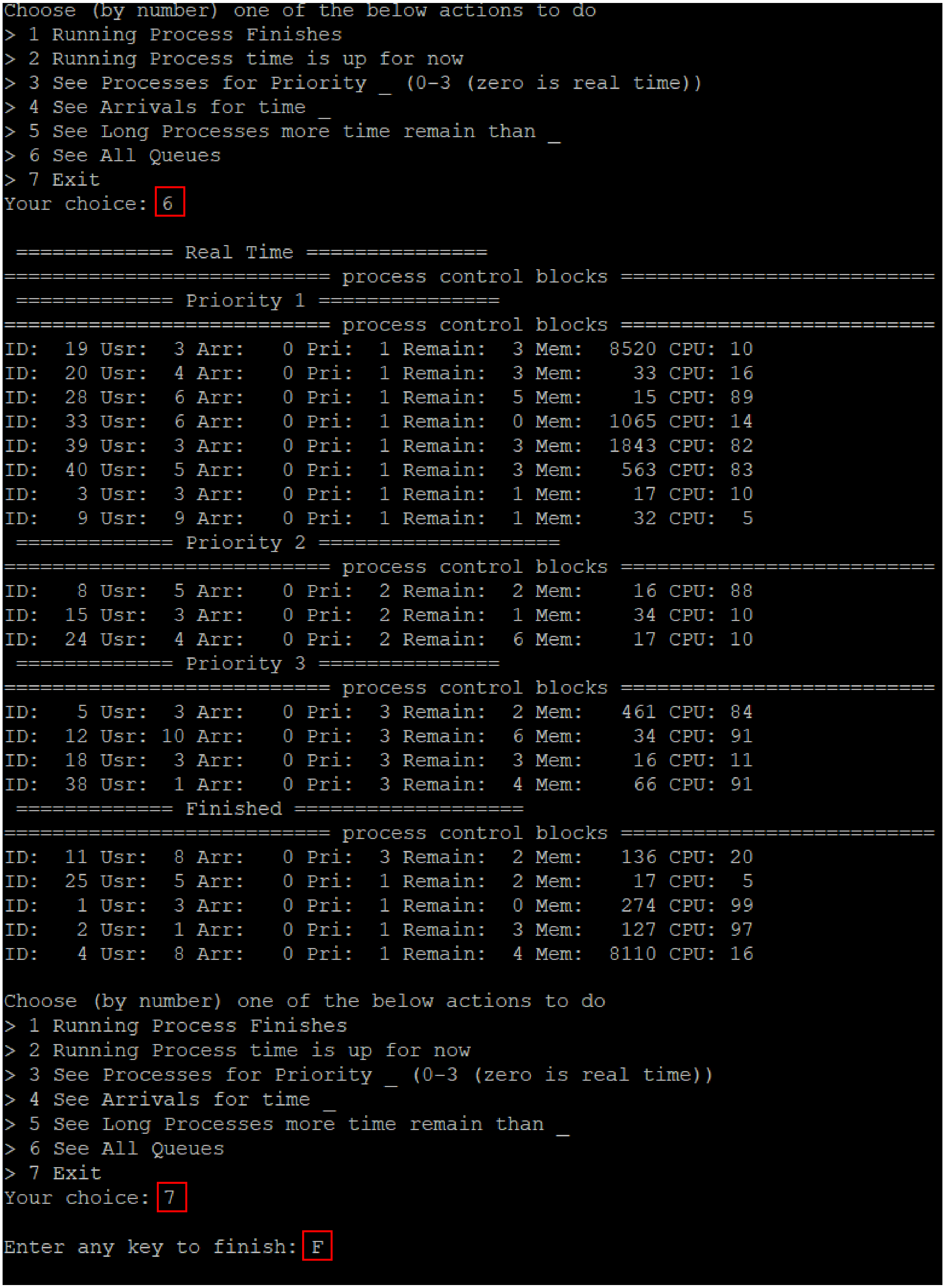
```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 1

Running:
ID:    9 Usr:   9 Arr:    0 Pri:   1 Remain:   1 Mem:     32 CPU:   5

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 2

Running:
ID:   14 Usr:   3 Arr:    0 Pri:   1 Remain:   2 Mem:     15 CPU:  10

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 3

What priority processes would you like to see? (0-3)
Please enter a value between 0 and 3 ==> 1
 ============= Priority 1 ===============
=========================== process control blocks ==========================
ID:   19 Usr:   3 Arr:    0 Pri:   1 Remain:   3 Mem:   8520 CPU:  10
ID:   20 Usr:   4 Arr:    0 Pri:   1 Remain:   3 Mem:     33 CPU:  16
ID:   28 Usr:   6 Arr:    0 Pri:   1 Remain:   5 Mem:     15 CPU:  89
ID:   33 Usr:   6 Arr:    0 Pri:   1 Remain:   0 Mem:   1065 CPU:  14
ID:   39 Usr:   3 Arr:    0 Pri:   1 Remain:   3 Mem:   1843 CPU:  82
ID:   40 Usr:   5 Arr:    0 Pri:   1 Remain:   3 Mem:    563 CPU:  83
ID:    3 Usr:   3 Arr:    0 Pri:   1 Remain:   1 Mem:     17 CPU:  10
ID:    9 Usr:   9 Arr:    0 Pri:   1 Remain:   1 Mem:     32 CPU:   5
```



```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 4

What simulated arrival time would you like to see processes from? (currently none higher than 100) 1
 ================= standby processes with arrival time = 1 === process control blocks ========================
ID:  22 Usr: 10 Arr:   1 Pri:  1 Remain:  1 Mem:    16 CPU: 94
     Number found: 1
```



```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 5

What time remaining would you like to see processes longer than? (currently low numbers) 3

== Real Time ====
 ================= estimated time remaining greater than 3 process control blocks ========================
     Number found: 0

== Priority 1 ====
 ================= estimated time remaining greater than 3 process control blocks ========================
ID:  28 Usr:  6 Arr:   0 Pri:  1 Remain:  5 Mem:    15 CPU: 89
     Number found: 1

== Priority 2 ====
 ================= estimated time remaining greater than 3 process control blocks ========================
ID:  24 Usr:  4 Arr:   0 Pri:  2 Remain:  6 Mem:    17 CPU: 10
     Number found: 1

== Priority 3 ====
 ================= estimated time remaining greater than 3 process control blocks ========================
ID:  12 Usr: 10 Arr:   0 Pri:  3 Remain:  6 Mem:    34 CPU: 91
ID:  38 Usr:  1 Arr:   0 Pri:  3 Remain:  4 Mem:    66 CPU: 91
     Number found: 2
```



```
Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 6

 ============= Real Time ===============
=========================== process control blocks ==========================
 ============= Priority 1 ===============
=========================== process control blocks ==========================
ID:  19 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  8520 CPU: 10
ID:  20 Usr:  4 Arr:   0 Pri:  1 Remain:  3 Mem:    33 CPU: 16
ID:  28 Usr:  6 Arr:   0 Pri:  1 Remain:  5 Mem:    15 CPU: 89
ID:  33 Usr:  6 Arr:   0 Pri:  1 Remain:  0 Mem:  1065 CPU: 14
ID:  39 Usr:  3 Arr:   0 Pri:  1 Remain:  3 Mem:  1843 CPU: 82
ID:  40 Usr:  5 Arr:   0 Pri:  1 Remain:  3 Mem:   563 CPU: 83
ID:   3 Usr:  3 Arr:   0 Pri:  1 Remain:  1 Mem:    17 CPU: 10
ID:   9 Usr:  9 Arr:   0 Pri:  1 Remain:  1 Mem:    32 CPU:  5
 ============= Priority 2 ====================
=========================== process control blocks ==========================
ID:   8 Usr:  5 Arr:   0 Pri:  2 Remain:  2 Mem:    16 CPU: 88
ID:  15 Usr:  3 Arr:   0 Pri:  2 Remain:  1 Mem:    34 CPU: 10
ID:  24 Usr:  4 Arr:   0 Pri:  2 Remain:  6 Mem:    17 CPU: 10
 ============= Priority 3 ===============
=========================== process control blocks ==========================
ID:   5 Usr:  3 Arr:   0 Pri:  3 Remain:  2 Mem:   461 CPU: 84
ID:  12 Usr: 10 Arr:   0 Pri:  3 Remain:  6 Mem:    34 CPU: 91
ID:  18 Usr:  3 Arr:   0 Pri:  3 Remain:  3 Mem:    16 CPU: 11
ID:  38 Usr:  1 Arr:   0 Pri:  3 Remain:  4 Mem:    66 CPU: 91
 ============= Finished ===================
=========================== process control blocks ==========================
ID:  11 Usr:  8 Arr:   0 Pri:  3 Remain:  2 Mem:   136 CPU: 20
ID:  25 Usr:  5 Arr:   0 Pri:  1 Remain:  2 Mem:    17 CPU:  5
ID:   1 Usr:  3 Arr:   0 Pri:  1 Remain:  0 Mem:   274 CPU: 99
ID:   2 Usr:  1 Arr:   0 Pri:  1 Remain:  3 Mem:   127 CPU: 97
ID:   4 Usr:  8 Arr:   0 Pri:  1 Remain:  4 Mem:  8110 CPU: 16

Choose (by number) one of the below actions to do
> 1 Running Process Finishes
> 2 Running Process time is up for now
> 3 See Processes for Priority _ (0-3 (zero is real time))
> 4 See Arrivals for time _
> 5 See Long Processes more time remain than _
> 6 See All Queues
> 7 Exit
Your choice: 7

Enter any key to finish: F
```

**Grading**

- Documentation and style 15%
- Correctness 80%
- Screenshots of program output 5%

**What you need to submit for this assignment:**

1. Your source code: **Assignment5.c** file.
2. A document in **Microsoft Word** or **PDF** format. It should include screenshots of your program output.

**When submitting on Canvas, please do not compress them into one file, but attach them one by one during the submission process.**