COMP 322/L—Introduction to Operating Systems and System Architecture
Assignment #2—Batch Process Scheduling

Objective:
> To calculate the timing parameters of batch processes based on different scheduling algorithms.

Specification:
> The program mimics the execution of different processes under different scheduling algorithms. The simulation maintains a table that reflects the current state of the system, based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:
> > 1) Enter parameters
> > 2) Schedule processes with the FIFO algorithm
> > 3) Schedule processes with the SJF algorithm
> > 4) Quit program and free memory

Assignment:
- Create a table to record the status of different processes based on the following parameters:
  - *id:* the unique identifier of the process
  - *arrival:* the point in time when the process enters the ready list to be executed by the CPU
  - *total_cpu:* the amount of CPU time the process will consume between arrival and departure
  - *done:* a flag that indicates whether the process has been successfully completed (1) or not (0)
  - *start_time:* the time when the process has begun being executed by the CPU
  - *end_time:* the time when the process has been completed by the CPU
  - *turnaround_time:* the sum of the total CPU time and the waiting time
- Calculate the values for the *start_time*, *end_time*, and *turnaround_time* for each process based on the selected scheduling algorithm.

What NOT to do (any violation will result in an automatic score of 0 on the assignment):
- Do NOT modify the choice values (1,2,3,4) or input characters and then try to convert them to integers--the test script used for grading your assignment will not work correctly.
- Do NOT turn in an alternate version of the assignment downloaded from the Internet (coursehero, chegg, reddit, github, etc.) or submitted from you or another student from a previous semester.
- Do NOT turn in your assignment coded in another programming language (C++, C#, Java).

What to turn in:
- The source code as a C file uploaded to Canvas by the deadline of 11:59pm PST (-20% per consecutive day for late submissions, up to the 4th day—note 1 minute late counts as a day late, 1 day and 1 minute late counts as 2 days late, etc.)
- Make sure your code compiles with the online C compiler before submitting: https://www.onlinegdb.com/online_c_compiler

Sample output

```
Batch scheduling
----------------
1) Enter parameters
2) Schedule processes with FIFO algorithm
3) Schedule processes with SJF algorithm
4) Quit program and free memory

Enter selection: 1
Enter total number of processes: 3
Enter process id: 1
Enter arrival time for process P[1]: 0
Enter total CPU time for process P[1]: 6
Enter process id: 2
Enter arrival time for process P[2]: 1
Enter total CPU time for process P[2]: 3
Enter process id: 3
Enter arrival time for process P[3]: 3
Enter total CPU time for process P[3]: 2

ID      Arrival Total   Start   End     Turnaround
--------------------------------------------------
1       0       6
2       1       3
3       3       2

Batch scheduling
----------------
1) Enter parameters
2) Schedule processes with FIFO algorithm
3) Schedule processes with SJF algorithm
4) Quit program and free memory

Enter selection: 2

ID      Arrival Total   Start   End     Turnaround
--------------------------------------------------
1       0       6       0       6       6
2       1       3       6       9       8
3       3       2       9       11      8

Batch scheduling
----------------
1) Enter parameters
2) Schedule processes with FIFO algorithm
3) Schedule processes with SJF algorithm
4) Quit program and free memory

Enter selection: 3

ID      Arrival Total   Start   End     Turnaround
--------------------------------------------------
3       3       2       6       8       5
2       1       3       8       11      10
1       0       6       0       6       6

Batch scheduling
----------------
1) Enter parameters
2) Schedule processes with FIFO algorithm
3) Schedule processes with SJF algorithm
4) Quit program and free memory

Enter selection: 4
Quitting program
```