Assignment 1

ADVANCE OPERATING SYSTEMS, FALL 2016

Time: One weeks

Instructions: Do not copy material from other sources, if it is necessary, then provide the references. In case of plagiarism, negative marks will be given to the plagiarizing person.

*Rules for assignment grading: The time for all assignments for the course of AOS will be 1 week. If there is at least one assignment submitted in time, then the time will exceed to 2 weeks. Of course, the late assignments will never have the full grade. If there is no assignment submission in one week, then there will be zero for all the class in that assignment. The grade A will only be given to a person who will submit **all** the assignments (no matter, in first week or second). The grade B will not be given to a person who has less than **75%** of submissions.*

There are two processes reading the same file named as `InputAssign-1.txt`. Each process reads the comma seperated file and prforms an action. The file looks like the following. The first row is the names of the processes. Each process will create a file same as the name given in the first row. For example in the file given here, there will be two files named `Process_A.txt` and `Process_B.txt`.

Process_A	Process_B
I am Process A	I am process B
[Hello From Proc A]	[Hello from Proc B]
[Message From Proc A]	Proc B is running
Proc A continuing	[Message from Proc B]

The process A (first process) will read the first row and output the the line "I am Process A" on the console and will write the same thing in its output file, i.e. `Process_A.txt`. Similarly, process B will do the same thing. For next line process A will read the string "[Hello from Proc A]", but because it is inside the brackets [ ], so process A will send this string to process B. Process B will output the string "[Hello from Proc A]" to the console and will also save it into the `Process_B.txt`. The same thing will happen at other end. The process B will read the string "[Hello from Proc B]" and send it to process A. Process A will you the string on console and also save it in `Process_A.txt`.

So as you can see the rule is, when a process reads a string without any brackets, then it outputs the string on its own console and stores it into the file. If it reads a string with brackets around it, then it sends it as a message to the other process. The other process receiving the message will output the message on its console and will also store it in the output file. If a process receives a message and it has to output a its own string, then its own string will be printed before the message. For example in 3rd row where process B has to output a message and its own string, its own string will be output before the message. So the output will be

Proc B is running
[Message From Proc A]

Note: If you cannot open different consoles for different processes, then its not a problem. The main thing is the correct output of the files.