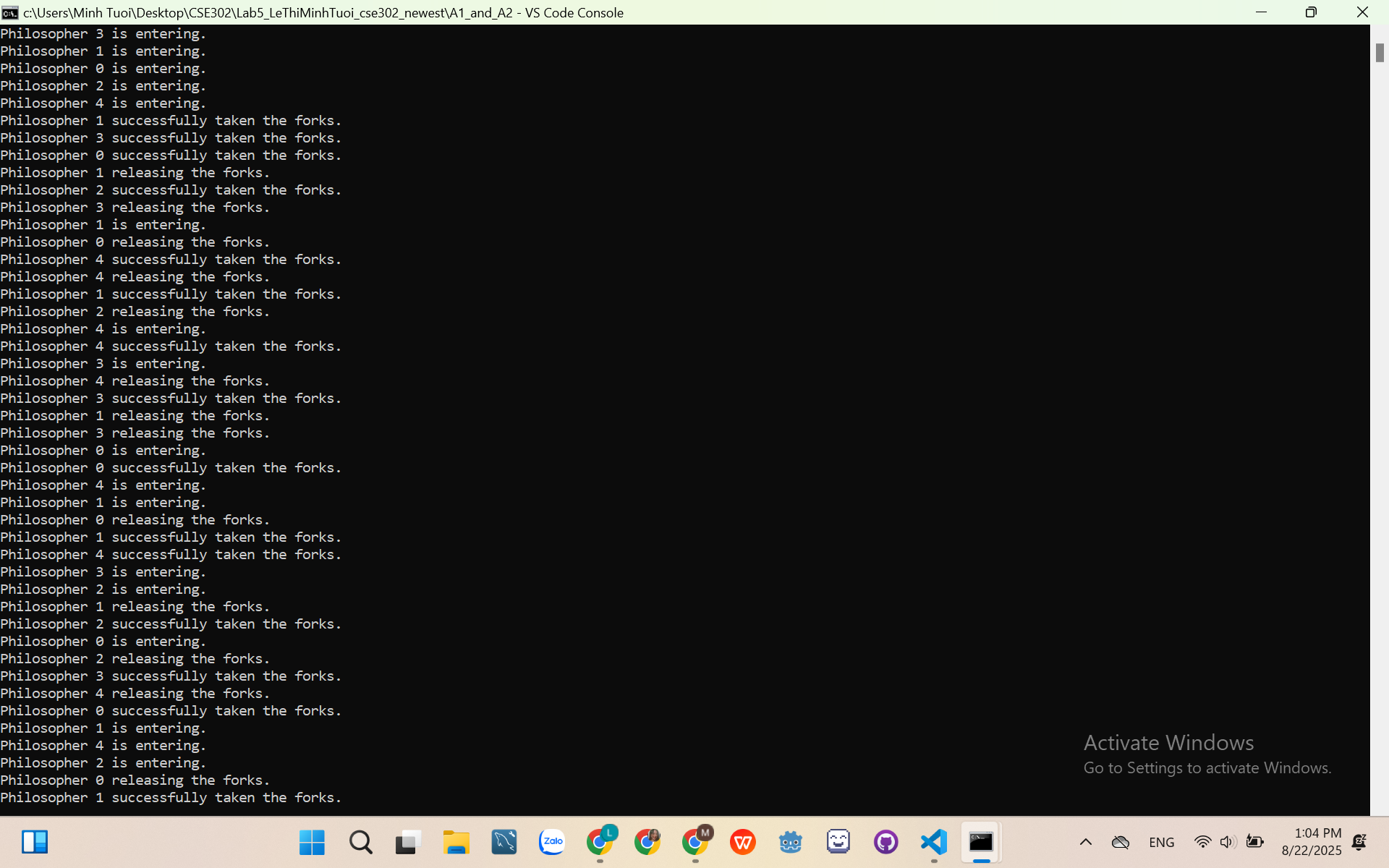
1>

Can we write a program for each philosopher that does what it is supposed to do and never gets stuck?

-> The program that I coded ensure deadlock-freedom but it does not ensure that starvation-freedom. If we want a starvation-free program, then( according to my search online ), we need coordinate with other methods to achieve that.

-> So the answer is yes, it is possible.



2>

Simple Rule:

First fork condition (canTakeFork()):

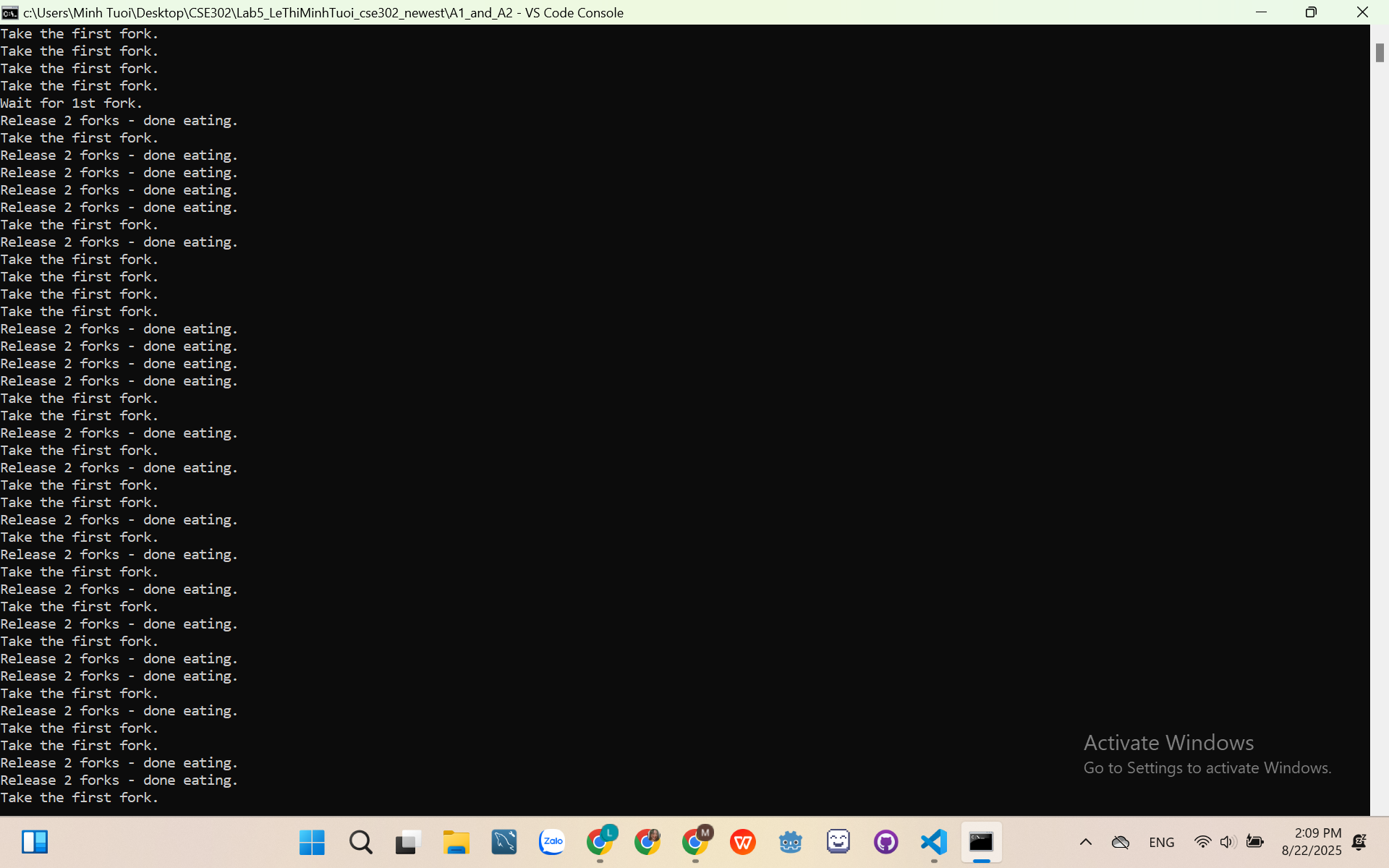
* If remainingForkNum > 1: safe to take one.
* If remainingForkNum == 1 && currentEaterNum > 0: someone is eating, so forks will be released soon.

Second fork condition:

* Wait until remainingForkNum > 0, then take it and start eating.

After eating:

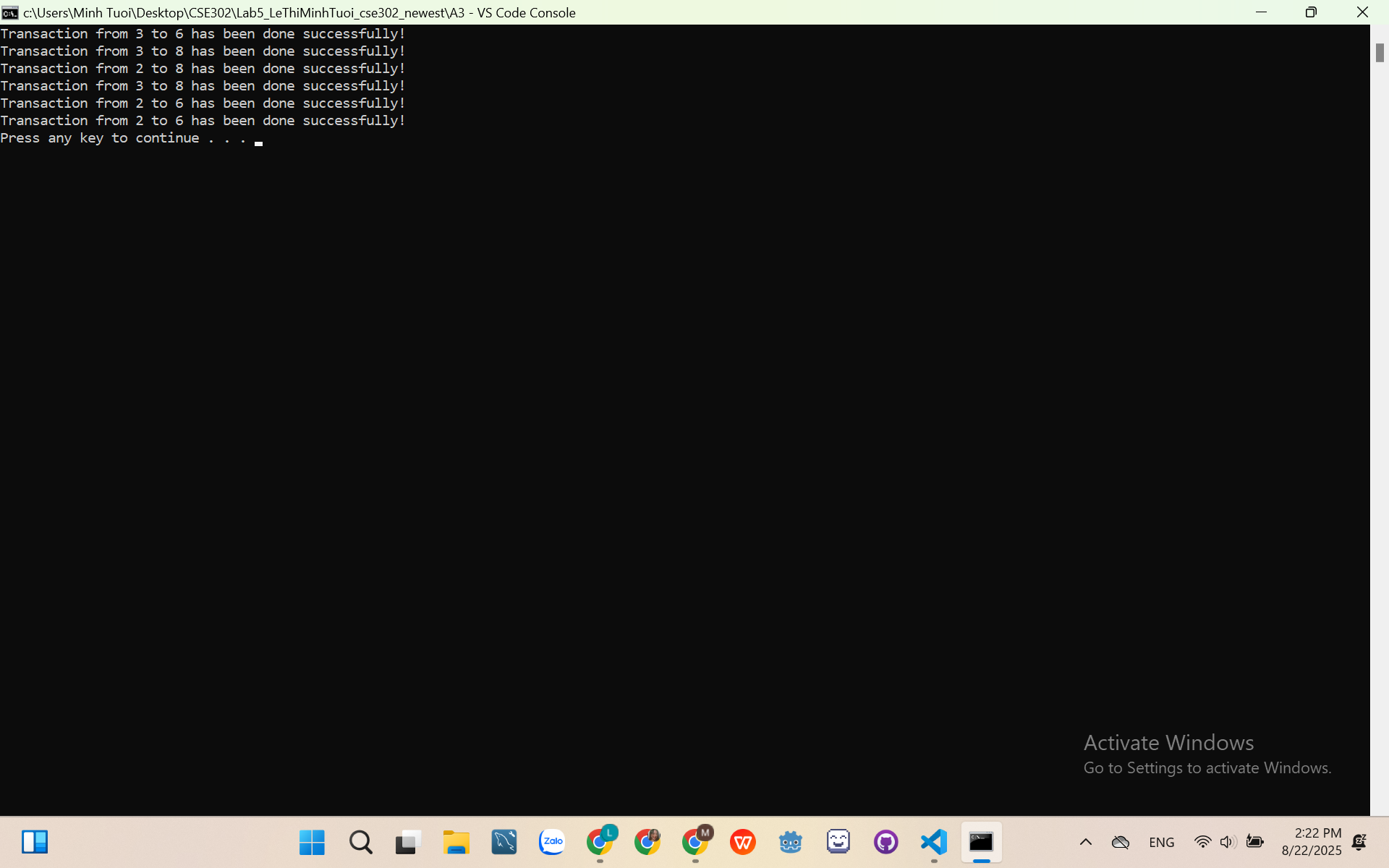
* Release both forks.
* Signal others waiting.



3>

Disadvantages of the sample program:

* If two accounts are transferring, then no other accounts can transfer until those two are done.
* If we remove the synchorized meaning many accounts can transfer at the same time, then deadlock can happen -> so we need to lock the accounts in order of their ID



4>

