**CS173**

**Project 1**

This project is a group project. You should complete the design and implementation of this project with your partner. Submit one program for the group.

Test your submission by running it on the linux machines in the linux lab (Olin 219 or Fellows 100). You must test it here first before submitting it. You can run your program on the linux boxes with

> python project1.py

As you review your design, consider features of your program such as elegance, simplicity, correctness, documentation, and testing for correctness. Your program must first be correct. Second, it should be elegant and efficient.

------------------------------------------------------------------------------------------------------------

You are to write a full python program to solve the following problem.

Meet Soprano the cat. Soprano wants to hop up on the kitchen counter, but he is not allowed to so he needs to be sneaky about it.



Two people live in Soprano’s house. They each cycle in and out of the kitchen following different but predictable schedules. At some point during the day, Soprano will attempt to hop up on the counter. If neither person is in the kitchen at the time of the hop, then Soprano will be successful. If either person is in the kitchen at the time of the hop, then Soprano will be caught.

Your program will first read two times (integers):

p1\_in length of time (in minutes) person 1 stays in the kitchen

p1\_out length of time (in minutes) person 1 stays out of the kitchen

Your program will then read two more times (integers):

p2\_in length of time (in minutes) person 2 stays in the kitchen

p2\_out length of time (in minutes) person 2 stays out of the kitchen

All times will be given in minutes and will be integer values >= 0. You can assume each day starts at time = 0 at midnight. The people who live in Soprano’s house both enter the kitchen at midnight, starting each of their “in” phases. At midnight each night each person starts a new cycle regardless of where they were in their prior cycles. They each remain in the kitchen for their respective times (p1\_in and p2\_in). Then they leave the kitchen for their respective times (p1\_out and p2\_out). They keep cycling in and out of the kitchen according to their time parameters until midnight the next night at which point they reset and begin a new series of cycles. You can assume that at least one of p\_in or p\_out is non-zero (the period cannot be 0).

Your program will also read the time of the escape:

t\_hop the time in minutes after midnight when Soprano attempts to hop

The t\_hop parameter is also an integer >= 0. Note: That makes five integer values read in by the program. They must be read in the order specified above. Do not change the order of input, nor should you prompt for anything other than these five integers.

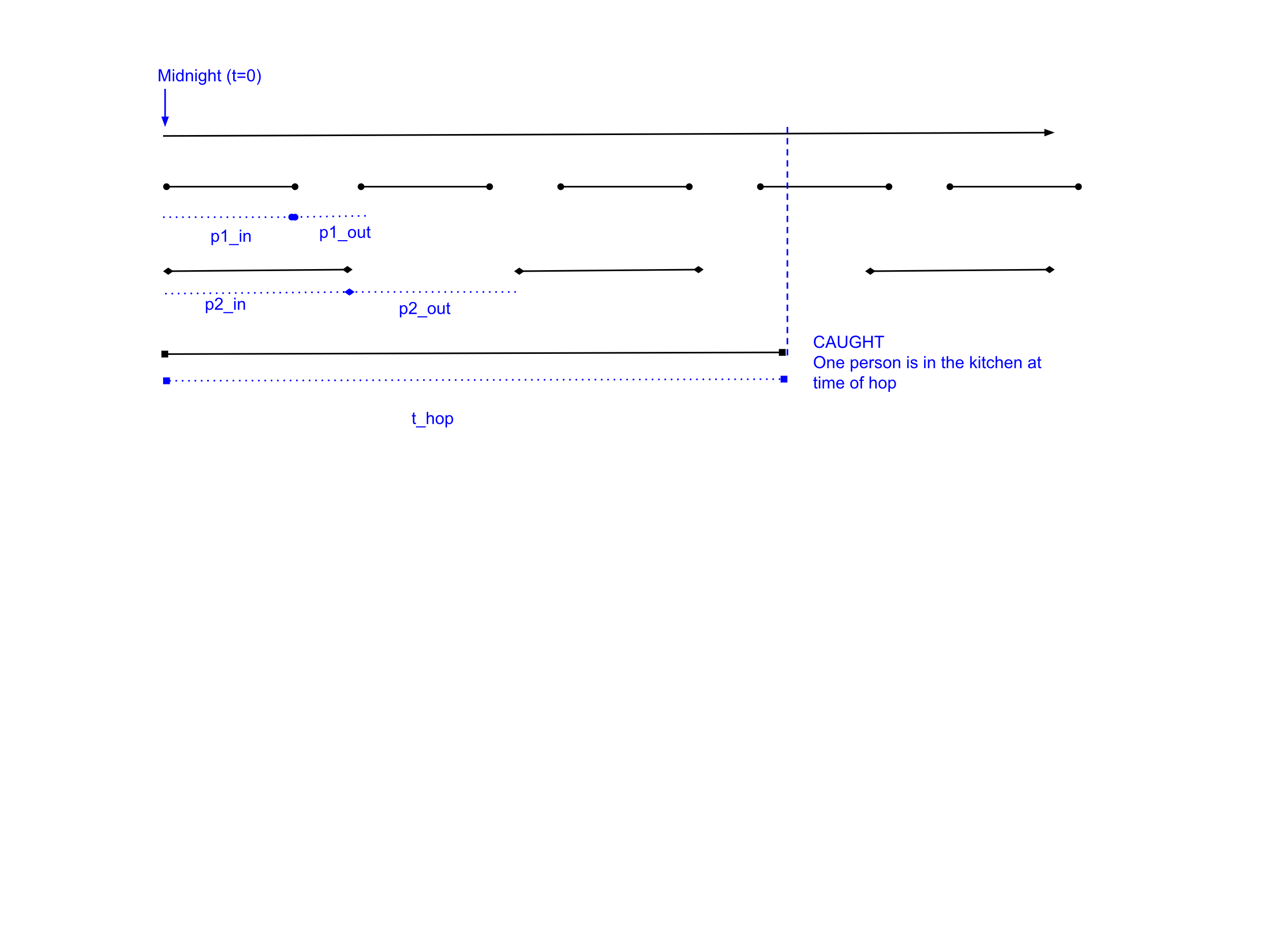
Your program should print one of the following responses:

SUCCESS if Soprano is successful in his hop attempt (no people in kitchen)

CAUGHT if Soprano is caught (one more more people in kitchen)

You may print extra information while you are debugging, but **your submitted solution should not print anything other than “SUCCESS” or “CAUGHT”** (i.e., do not print a prompt saying, “Enter p1\_in:” – just accept the input assuming the user knows which values to provide in which order).

An example is shown below in which one of the two people (person 1) is in the kitchen when Soprano tries to hop up. A person is considered "out" if the hop happens just at the end of the person's in-phase and considered "in" if the escape happens just at the start of the person's in-phase.



All input values are integers greater than or equal to 0 (with a non zero period). Make no other assumptions about the input. Part of this assignment will be for your team to think hard about special cases of input and how to handle them.

Levels of achievement:

* D range: A non-working (but still well-documented) program. May have runtime errors but points will be given based on progress toward a working solution.
* C range: A well-documented program that works for common cases (in the middle of in and out cycles).
* B range: A well-documented program that works for common and edge cases.
* A range: A working, well-documented program that does not use loops or other unnecessary iteration (the most elegant solution to this problem can be done with no iteration).