CSE 532 Homework Assignment 1

The Golem and the Imp

A lazy wizard has summoned an imp into his tower to perform chores and menial tasks. The imp finds this arrangement boring and wishes to escape the tower for more entertaining pursuits. Unfortunately for it, the wizard has animated a mechanical construct (golem) to keep an eye on the imp. In a lucky turn of events, the imp has located a secret trap door leading into the cellar tunnels, which eventually leads to an escape to the surface outside the tower. The bad news is that the golem is right on its heels. One of two events is about to transpire: either the imp is going to escape the clutches of the wizard to sweet open-air and freedom, or it is going to be recaptured and return to a dull life in the tower.

1. **(70 pts) You are to create an IPython notebook file* entitled GISimulation.ipynb with an obvious cell that performs a simulation of the imp-golem chase to the surface. Assume the following rules:**
   i. The tunnel is 50 feet in length (beginning at foot 0, ending at foot 50). The golem starts at the beginning (foot 0), while the imp has a short head-start (at foot 5).
   ii. While in the tunnel, the imp moves a bit awkwardly between 1 and 9 feet (randomly) every second, while the golem moves at a slow but steady 3-5 feet (randomly) every second.
   iii. If the imp reaches foot 50 (the exit to the surface) it can use its wings to easily escape. But if the golem's position overtakes (>=) the imp's position beforehand, the imp is captured and its fun ends.
   iv. You are to simulate every second of the chase by incrementing the golem's and imp's position according to the rules in part ii. above, culminating with one of the scenarios described in part iii. You must print details regarding each second's passing (where the golem and imp are), up until the concluding second – at which point you should simply print the outcome. (See examples below.)

Ex 1 (The imp escapes):

```
G----I-----------------------------------------X       0s
----G-------I-----------------------------------X       1s
--------G---------I-----------------------------X       2s
----------G----------I--------------------------X       3s
--------------G----------I----------------------X       4s
----------------G-----------I--------------------X       5s
--------------------G-----------I---------------X       6s
------------------------G-------------I--------X       7s
----------------------------G----------------I--X       8s
THE IMP HAS ESCAPED TO FREEDOM AFTER 9 SECONDS.  MISCHIEF AND TRICKERY AWA
IT!
```

Ex 2 (The golem captures the imp):

```
G----I-----------------------------------------X       0s
----G---I---------------------------------------X       1s
-------G---I-------------------------------------X       2s
-----------G--I-----------------------------------X       3s
---------------G-I--------------------------------X       4s
SADLY, IT'S BACK TO THE TOWER FOR THE IMP AFTER A 5 SECOND-CHASE.
```

**Note that you can use a simple .py script (GISimulation.py), but your code must be readable and you must take steps to make sure your chase is presentable (i.e. output comes across like the scenarios above). A .py script that is difficult to use or read will be penalized!*

2. **(30 pts) You are to define Python function GISimple(…) within a standard python file GISimple.py that simulates the chase from problem 1 but prints nothing and instead returns *True* (if the imp escapes) or *False* (if it is recaptured). However, instead of using fixed positions and speeds, your function must accept the following parameters:**
   1. impSpd: A two-element tuple where element 1 is the min # feet per second and element 2 is the max # for the imp (default of (1,9)).
   2. golemSpd: A two-element tuple identical to impSpd, but gives the golem's movement in feet per second (default of (3,5)).
   3. headStart: a single integer indicating the initial position of the imp in feet (default of 5).
   4. exitPosition: A single integer indicating the foot at which the exit appears (default of 50).

   Make sure you include the default values in your function definition!

   The parameters must be passed to function GISimple in the above order. Its signature should therefore be GISimple(impSpd, golemSpd, headStart, exitPosition). The actual "chase" is identical to that of problem 1, but should substitute the parameters in place of the fixed values when applying rules.

   Note that we will be testing your function with a script of our own (via an import statement) when grading your code. To that end, it is highly advised that you test your function works correctly by calling it from an outside script, or importing it within a ipynb file and testing it.

   Your submission to Blackboard should be a single .zip file with the name <LN>_<FN>_1.zip, where <LN> and <FN> are your last name and first name respectively. The file should contain GISimulation.ipynb (or GISimulation.py) and GISimple.py, optionally with a README file for documentation.