Part B - Problem 5 | Problem Set 3 | Contenu du cours 6.00.2x

In this problem, we will use the implementation you filled in for Problem 4 to run a simulation. You will create a `TreatedPatient` instance with the following parameters, then run the simulation:

- `viruses`, a list of 100 `ResistantVirus` instances
- `maxPop`, maximum sustainable virus population = 1000

Each `ResistantVirus` instance in the viruses list should be initialized with the following parameters:

- `maxBirthProb`, maximum reproduction probability for a virus particle = 0.1
- `clearProb`, maximum clearance probability for a virus particle = 0.05
- `resistances`, The virus's genetic resistance to drugs in the experiment = {'guttagonol': False}
- `mutProb`, probability of a mutation in a virus particle's offspring = 0.005

Run a simulation that consists of 150 time steps, followed by the addition of the drug, guttagonol, followed by another 150 time steps. You should make use of the function `simulationWithDrug(numViruses, maxPop, maxBirthProb, clearProb, resistances, mutProb, numTrials)`. As with problem 3, perform up to 100 trials and make sure that your results are repeatable and representative.

Create one plot that records both the average total virus population and the average population of guttagonol-resistant virus particles over time.

A few good questions to consider as you look at your plots are: What trends do you observe? Are the trends consistent with your intuition? Feel free to discuss the answers to these questions in the forum, to fully cement your understanding of this problem set, processing and interpreting data.

Again, as in Problem 3, you can use the provided .pyc file to check that your implementation of the `TreatedPatient` and `ResistantVirus` classes work as expected.