PART B - PROBLEM 5: RUNNING AND ANALYZING A SIMULATION WITH A DRUG (10/10 points)

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.

```
1  # Enter your definition for simulationWithDrug in this box
2  #
3  # PROBLEM 5
4  #
5  def simulationWithDrug(numViruses, maxPop, maxBirthProb, clearProb, resistances,
6                    mutProb, numTrials):
7      """
8      Runs simulations and plots graphs for problem 5.
9
10     For each of numTrials trials, instantiates a patient, runs a simulation for
11     150 timesteps, adds guttagonol, and runs the simulation for an additional
12     150 timesteps.  At the end plots the average virus population size
13     (for both the total virus population and the guttagonol-resistant virus
14     population) as a function of time.
15
16     numViruses: number of ResistantVirus to create for patient (an integer)
```

Correct

Test results

CORRECT

Test: simulation 1

Note: your simulation should only be using the plot, title, xlabel, ylabel, legend, figure, and show functions from the Pylab module.

For grading, please plot the total viruses first, then plot the resistant viruses second, in two separate calls to 'pylab.plot.'

Be sure you call pylab.show() last. The order of other lines do not matter.

Output:

```
Test: simulationWithDrug(1, 10, 1.0, 0.0, {}, 1.0, 5)
Plotting...
Plotting values: [2.0, 3.8, 6.4, 8.0, 9.4, 10.0, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
10.2, 10.2]
Plotting...
Plotting values: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
Successfully called pylab.xlabel with label: time step
Successfully called pylab.ylabel with label: # viruses
Successfully called pylab.title with label: ResistantVirus simulation
Successfully called pylab.legend
Successfully called pylab.show
```

Test: simulation 2

Note: your simulation should only be using the plot, title, xlabel, ylabel, legend, figure, and show functions from the Pylab module.

For grading, please plot the total viruses first, then plot the resistant viruses second, in two separate calls to 'pylab.plot.'

Be sure you call pylab.show() last. The order of other lines do not matter.

Output:

```
Test: simulationWithDrug(1, 20, 1.0, 0.0, {"guttagonol": True}, 1.0, 5)
Plotting...
Plotting values: [2.0, 4.0, 7.4, 12.6, 17.0, 19.8, 20.0, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2, 20.2,
 20.2, 20.2]
Plotting...
Plotting values: [1.0, 2.0, 3.8, 6.4, 8.2, 9.8, 10.0, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2, 10.2,
 10.2, 10.2]
Successfully called pylab.xlabel with label: time step
Successfully called pylab.ylabel with label: # viruses
Successfully called pylab.title with label: ResistantVirus simulation
Successfully called pylab.legend
Successfully called pylab.show
```

Test: simulation 3

Note: your simulation should only be using the plot, title, xlabel, ylabel, legend, figure, and show functions from the Pylab module.

For grading, please plot the total viruses first, then plot the resistant viruses second, in two separate calls to 'pylab.plot.'

Be sure you call pylab.show() last. The order of other lines do not matter.

Output:

```
Test: simulationWithDrug(75, 100, .8, 0.1, {"guttagonol": True}, 0.8, 1)
Plotting...
Plotting values: [80.0, 90.0, 98.0, 98.0, 97.0, 96.0, 94.0, 93.0, 96.0, 99.0, 97.0, 96.0,
99.0, 95.0, 96.0, 96.0, 91.0, 94.0, 90.0, 91.0, 97.0, 95.0, 92.0, 89.0, 86.0, 89.0, 94.0,
96.0, 98.0, 98.0, 98.0, 98.0, 97.0, 99.0, 100.0, 96.0, 98.0, 96.0, 100.0, 97.0, 97.0, 97.0,
97.0, 96.0, 96.0, 98.0, 95.0, 98.0, 96.0, 99.0, 95.0, 89.0, 92.0, 89.0, 88.0, 94.0, 93.0,
96.0, 94.0, 98.0, 97.0, 94.0, 98.0, 96.0, 98.0, 97.0, 97.0, 100.0, 96.0, 99.0, 100.0, 100.0,
100.0, 96.0, 95.0, 99.0, 99.0, 92.0, 89.0, 97.0, 104.0, 98.0, 96.0, 98.0, 95.0, 99.0, 100.0,
98.0, 96.0, 99.0, 97.0, 98.0, 99.0, 100.0, 97.0, 95.0, 97.0, 102.0, 101.0, 93.0, 94.0, 95.0,
95.0, 98.0, 98.0, 97.0, 99.0, 99.0, 91.0, 100.0, 98.0, 95.0, 93.0, 97.0, 95.0, 94.0, 96.0,
100.0, 100.0, 100.0, 97.0, 96.0, 93.0, 94.0, 89.0, 88.0, 96.0, 95.0, 94.0, 96.0, 99.0, 94.0,
92.0, 87.0, 93.0, 99.0, 99.0, 99.0, 95.0, 96.0, 97.0, 94.0, 99.0, 93.0, 95.0, 97.0, 101.0,
100.0, 97.0, 91.0, 90.0, 86.0, 86.0, 82.0, 85.0, 88.0, 84.0, 86.0, 84.0, 90.0, 88.0, 85.0,
83.0, 87.0, 83.0, 77.0, 74.0, 77.0, 75.0, 75.0, 69.0, 69.0, 72.0, 74.0, 65.0, 60.0, 58.0,
58.0, 56.0, 55.0, 55.0, 54.0, 51.0, 53.0, 52.0, 43.0, 47.0, 49.0, 50.0, 53.0, 58.0, 56.0,
51.0, 50.0, 54.0, 51.0, 48.0, 51.0, 54.0, 51.0, 49.0, 46.0, 45.0, 49.0, 50.0, 53.0, 55.0,
59.0, 62.0, 60.0, 61.0, 60.0, 54.0, 54.0, 54.0, 57.0, 53.0, 51.0, 48.0, 45.0, 42.0, 41.0,
41.0, 45.0, 48.0, 44.0, 50.0, 48.0, 50.0, 51.0, 49.0, 46.0, 46.0, 47.0, 44.0, 43.0, 43.0,
41.0, 38.0, 36.0, 33.0, 37.0, 35.0, 32.0, 31.0, 36.0, 36.0, 35.0, 35.0, 36.0, 34.0, 33.0,
29.0, 29.0, 28.0, 28.0, 32.0, 34.0, 31.0, 33.0, 31.0, 33.0, 35.0, 33.0, 35.0, 36.0, 31.0,
35.0, 40.0, 39.0, 40.0, 40.0, 39.0, 38.0, 41.0, 36.0, 40.0, 43.0, 47.0, 52.0, 51.0, 51.0,
55.0, 52.0, 52.0, 52.0, 56.0, 55.0, 58.0, 57.0, 52.0, 50.0, 48.0, 43.0, 38.0, 38.0, 34.0,
32.0, 29.0, 28.0]
Plotting...
Plotting values: [68.0, 68.0, 65.0, 62.0, 64.0, 60.0, 55.0, 54.0, 53.0, 48.0, 46.0, 48.0,
47.0, 44.0, 45.0, 47.0, 47.0, 49.0, 48.0, 50.0, 56.0, 54.0, 55.0, 54.0, 50.0, 47.0, 48.0,
48.0, 50.0, 47.0, 46.0, 42.0, 40.0, 40.0, 41.0, 42.0, 46.0, 43.0, 44.0, 41.0, 44.0, 42.0,
43.0, 46.0, 49.0, 51.0, 48.0, 52.0, 47.0, 49.0, 45.0, 40.0, 43.0, 42.0, 40.0, 42.0, 42.0,
42.0, 44.0, 45.0, 43.0, 41.0, 45.0, 42.0, 46.0, 44.0, 43.0, 44.0, 40.0, 44.0, 42.0, 42.0,
42.0, 43.0, 40.0, 43.0, 43.0, 40.0, 39.0, 43.0, 46.0, 44.0, 41.0, 44.0, 44.0, 46.0, 48.0,
46.0, 43.0, 45.0, 43.0, 44.0, 48.0, 48.0, 51.0, 52.0, 53.0, 50.0, 50.0, 46.0, 48.0, 48.0,
47.0, 51.0, 48.0, 49.0, 44.0, 43.0, 41.0, 53.0, 51.0, 51.0, 50.0, 52.0, 51.0, 49.0, 46.0,
47.0, 45.0, 44.0, 44.0, 44.0, 42.0, 43.0, 43.0, 41.0, 44.0, 45.0, 43.0, 46.0, 50.0, 48.0,
44.0, 42.0, 44.0, 50.0, 49.0, 51.0, 49.0, 50.0, 47.0, 41.0, 46.0, 38.0, 40.0, 44.0, 49.0,
47.0, 42.0, 44.0, 43.0, 42.0, 39.0, 36.0, 37.0, 37.0, 34.0, 35.0, 33.0, 33.0, 33.0, 33.0,
32.0, 31.0, 29.0, 25.0, 24.0, 25.0, 26.0, 24.0, 23.0, 21.0, 22.0, 25.0, 15.0, 15.0, 15.0,
15.0, 11.0, 10.0, 11.0, 11.0, 13.0, 14.0, 14.0, 11.0, 12.0, 12.0, 10.0, 11.0, 12.0, 12.0,
11.0, 12.0, 14.0, 13.0, 13.0, 14.0, 16.0, 15.0, 15.0, 14.0, 14.0, 14.0, 17.0, 16.0, 16.0,
18.0, 17.0, 18.0, 16.0, 16.0, 13.0, 13.0, 12.0, 13.0, 13.0, 14.0, 12.0, 10.0, 10.0, 10.0,
10.0, 12.0, 12.0, 10.0, 12.0, 13.0, 15.0, 14.0, 14.0, 11.0, 10.0, 10.0, 9.0, 10.0, 10.0,
11.0, 9.0, 9.0, 8.0, 7.0, 7.0, 7.0, 6.0, 7.0, 7.0, 6.0, 5.0, 7.0, 6.0, 6.0, 5.0, 6.0, 7.0,
8.0, 8.0, 11.0, 7.0, 8.0, 6.0, 6.0, 5.0, 6.0, 7.0, 10.0, 9.0, 10.0, 11.0, 8.0, 7.0, 7.0, 7.0,
7.0, 9.0, 9.0, 11.0, 12.0, 11.0, 14.0, 14.0, 15.0, 15.0, 14.0, 13.0, 13.0, 13.0, 12.0, 12.0,
10.0, 10.0, 7.0, 6.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
Successfully called pylab.xlabel with label: time step
Successfully called pylab.ylabel with label: # viruses
Successfully called pylab.title with label: ResistantVirus simulation
Successfully called pylab.legend
Successfully called pylab.show
```

Hide output

Check   Save   *You have used 5 of 30 submissions*

Show Discussion   New Post

edX

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+