CISC106 Summer 2011 Lab06

- This lab an all subsequent labs will be due Sunday at 11:55 PM EDT on Sakai.

Preparation (do not submit for grading)

1. Start up the python interpreter and `import random`. Call `random.seed`(490177). Then call `random.randint`$(0, 1000000)$ several times and see what numbers you get.

2. Call `random.seed`(490177) again. Then go back to calling `random.randint`$(0, 1000000)$. What do you notice? What happens if you call `random.seed` with another seed besides 490177?

Programs (to be graded)

1. The *Monty Hall Problem* is a probability puzzle named after the host of the 1960s/70s game show *Let's Make a Deal*. The problem statement is as such:[1]

   > You're a contestant on a gameshow. Before you are three doors - aptly labeled 1, 2 and 3. The show's host explains to you that you that behind *one* of these doors is a car, and that a goat resides behind each of the other two. You must choose one of these three doors, at which point the host will open one of the other two doors revealing its contents. You will then be given the option to switch to other un-opened door. Whether or not you choose to switch, if the door you choose contains the car, you win that car. If it contains a goat, you win nothing. You choose your door, and then the hosts opens one of the other two doors to reveal a goat. It's now time to decide what to do next - should you stay or should you go? Which choice, if either, gives you a better chance of winning the car?

   Your job is to write a simple simulation of one play of the game. In the simulation, you should randomly generate the contestant's choice of doors *and* the door containing the car. You can do this by importing the `random` package and then calling `random.randint`$(1, 3)$ - which will generate a random number between 1 and 3 (inclusive.) You should then consider what the win conditions are for a contestant who chooses to switch and for one who chooses *not* to switch. When thinking about these conditions, keep in mind the fact that *the host is aware of which door contains the car* when he performs the reveal.

2. Next, you should write a function `run_test`, which takes a number *iterations* and a boolean specifying whether or not to switch doors. It will then run your simulation *iterations* times, counting the number of times the contestant wins the car. After running the simulations, it should display the raw number of wins as well as the win ratio. Try calling `run_test` several times, varying whether or not to switch, to see which strategy wins you the car more often.[2]

You should implemnt your simulation in a file called lab06.py. Submit that file along with any other docs required by your TA on Sakai.

[1] For more info on the Monty Hall Problem, visit the wikipedia article: `http://en.wikipedia.org/wiki/Monty_Hall_problem`

[2] One final note - be sure to intialize the random number generator before calling `run_test` by calling `random.seed()`