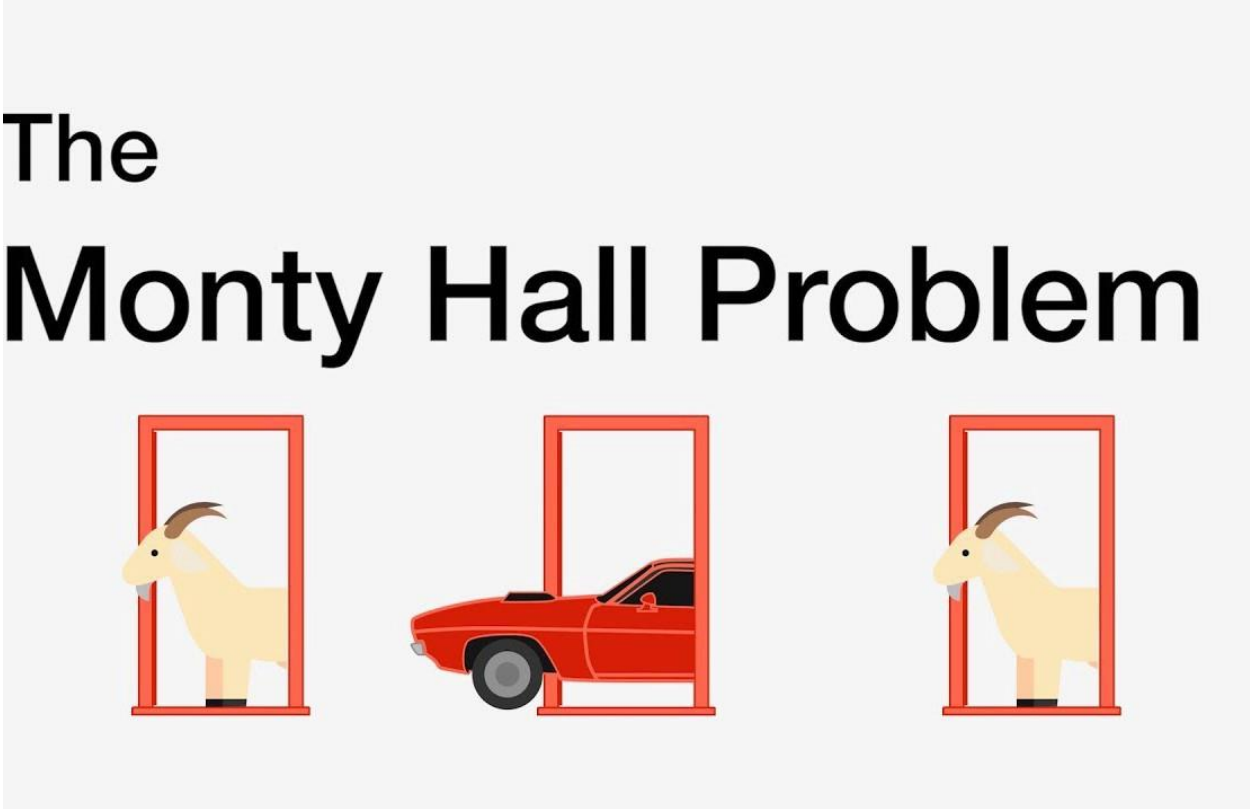
Name: Daniel Thommana Python ASSGN 2

1)



The Monty Hall problem is a well-known puzzle in probability derived from an American game show, Let's Make a Deal. (The original 1960s-era show was hosted by Monty Hall, giving this puzzle its name.) Intuition leads many people to get the puzzle wrong, and when the Monty Hall problem is presented in a newspaper or discussion list, it often leads to a lengthy argument in letters-to-the-editor and on message boards.

The game is played like this:

* The game show set has three doors. A prize such as a car or vacation is behind one door, and the other two doors hide a valueless prize called a Zonk; in most discussions of the problem, the Zonk is a goat.

* The contestant chooses one door. We'll assume the contestant has no inside knowledge of which door holds the prize, so the contestant will just make a random choice.

* The smiling host Monty Hall opens one of the other doors, always choosing one that shows a goat, and always offers the contestant a chance to switch their choice to the remaining unopened door.

The contestant either chooses to switch doors, or opts to stick with the first choice.

* Monty calls for the remaining two doors to open, and the contestant wins whatever is behind their chosen door.

* Let's say a hypothetical contestant chooses door #2. Monty might then open door #1 and offer the chance to switch to door #3. The contestant switches to door #3, and then we see if the prize is behind #3.

The puzzle is to identify if switching increases the chance of winning the car, decreases it, or makes no difference? Use python code to play this game multiple times (1000) and analyse your observations ?

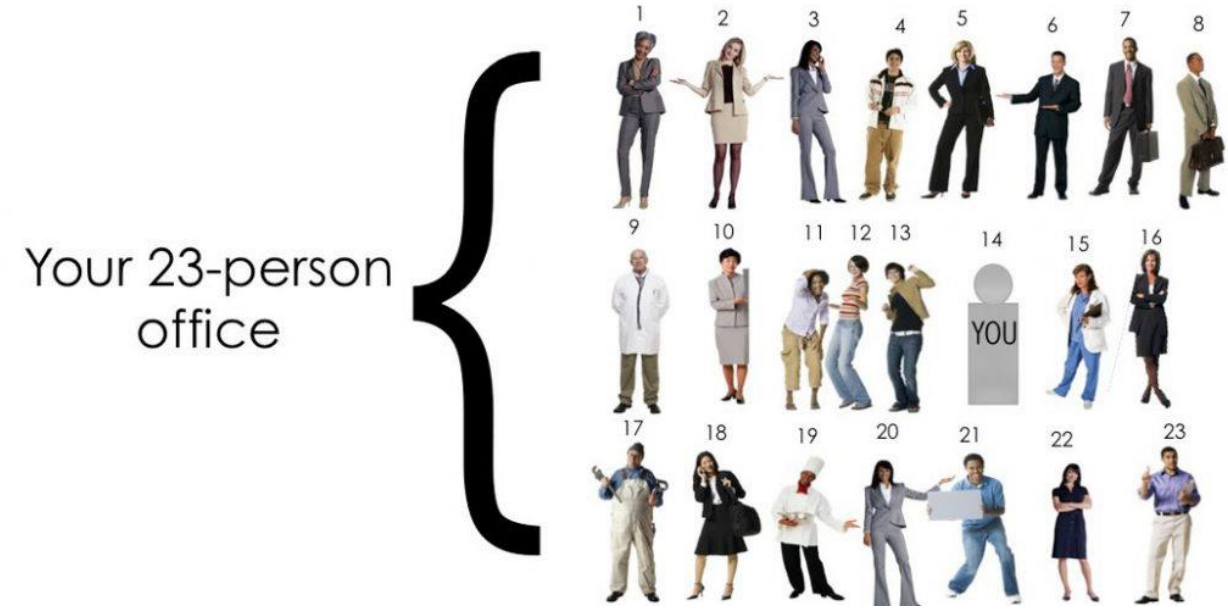
Ans: You should **switch** doors to have a **66%** chance of winning

Probability of winning doubles when you switch doors.

```
import random

def run_trial(switch_doors, ndoors=3):


    # Pick a random door out of the ndoors available
    chosen_door = random.randint(1, ndoors)
    if switch_doors:
        # Reveal a goat
        revealed_door = 3 if chosen_door==2 else 2
        # Make the switch by choosing any other door than the initially-
        # selected one and the one just opened to reveal a goat.
        available_doors = [dnum for dnum in range(1,ndoors+1)
                                if dnum not in (chosen_door, revealed_door
)]
        chosen_door = random.choice(available_doors)

    # You win if you picked door number 1
    return chosen_door == 1

def run_trials(ntrials, switch_doors, ndoors=3):


    nwins = 0
    for i in range(ntrials):
        if run_trial(switch_doors, ndoors):
            nwins += 1
    return nwins

ndoors, ntrials = 3, 1000
nwins_without_switch = run_trials(ntrials, False, ndoors)
nwins_with_switch = run_trials(ntrials, True, ndoors)

print('Monty Hall Problem with {} doors'.format(ndoors))
print('Proportion of wins without switching: {:.4f}'
            .format(nwins_without_switch/ntrials))
print('Proportion of wins with switching: {:.4f}'
            .format(nwins_with_switch/ntrials))
```

```
Monty Hall Problem with 3 doors
Proportion of wins without switching: 0.3360
Proportion of wins with switching: 0.6640
```

2)



Let's say there are 23 employees in any office party. What is the probability that at least two of the employees will have the same birthday ?

Ans: 0.50

```
import math

def nCr(n,r):
    f = math.factorial
    return f(n) / f(r) / f(n-r)


one_person_birthday = 1/365
two_person_birthday = 364/365
prob = 1-pow(two_person_birthday,nCr(23,2))
print(prob)
```

```
0.5004771540365807
```