```
import numpy as np
import random
```

```
goats = ['first goat', 'second goat']
hidden_behind_doors = np.append(goats, 'car')
print(hidden_behind_doors)
```

```
['first goat' 'second goat' 'car']
```

```
# Playing the game where contestant always switches doors
def MontyHallSwitch(numGames):
  switch_wins = 0
  for _ in range(numGames):
    random.shuffle(hidden_behind_doors)   # shuffle car & goats location behind doors
    game_layout = hidden_behind_doors     # save shuffled doors as layout for this game

    contestant_choice = random.randint(0,2)           # random pick door number for contestant
    contestant_door = game_layout[contestant_choice]  # determine what is behind contestant door,
                                                      # but contestant doesn't know
    if contestant_door == 'car':
      switch_wins += 0    # lose by switching

    else:                 # contestant door is either 'first goat' or 'second goat'
      switch_wins += 1    # win by switching

  switch_win_percentage = switch_wins / numGames * 100

  print('In {} Simulations...\nSwitch Original Door Wins: {}%'.format(numGames, switch_win_percentage))

# Playing the game where contestant always keeps original door
def MontyHallKeep(numGames):
  keep_wins = 0
  for _ in range(numGames):
    random.shuffle(hidden_behind_doors)   # shuffle car & goats location behind doors
    game_layout = hidden_behind_doors     # save shuffled doors as layout for this game

    contestant_choice = random.randint(0,2)           # random pick door number for contestant
    contestant_door = game_layout[contestant_choice]  # determine what is behind contestant door,
                                                      # but contestant doesn't know
    if contestant_door == 'car':
      keep_wins += 1    # win by keeping

    else:               # contestant door is either 'first goat' or 'second goat'
      keep_wins += 0    # lose by keeping

  keep_win_percentage = keep_wins / numGames * 100

  print('In {} Simulations...\nKeep Original Door Wins: {}%'.format(numGames, keep_win_percentage))
```

```
MontyHallKeep(50000)
print()
MontyHallSwitch(50000)
```

```
In 50000 Simulations...
Keep Original Door Wins: 33.47%

In 50000 Simulations...
Switch Original Door Wins: 66.374%
```