Artificial Intelligence  
Lab Exercise 7  
Monty Hall Problem

short line

Ashwin Prakash  
RA1911026010048

**Monty Hall Problem:**

The Monty Hall problem is a counter – intuitive statistics puzzle:

* There are 3 doors, which are two goats and a car
* You pick a door hoping it to be a car
* The game show host, examines the other doors and opens one with a goat. (If both doors have goats, he picks randomly.)

**Aim:**

To implement Monty Hall Problem

**Problem Statement:**

To implement the Monty Hal problem through an interactive game played by the user in the terminal window

**Algorithm:**

1. Generate random environments (trials) to guess, and visualize them using Matplolib/Seaborn.
2. Create Stay, switch and door containers.
3. Randomly assign door with a car, and randomly assign a guess for contestant.
4. Host reveals a goat, The host has 2 doors to choose from [contestant\_guess == car - door]. If host has only 1 door, then [contestant\_guess != car - door]
5. Define the door that the contestant would switch to
6. Record and display the result of switching or staying outcomes

**Program:**

import random

def get\_non\_prize\_door(host, num\_doors, player\_choice):

  i = 1

  while (i == host or i== player\_choice ):

    i = (i+1)%(num\_doors)

  return i

def switch\_function(shown\_door, num\_doors, player\_choice):

  i = 1

  while (i == shown\_door or i== player\_choice ):

    i = (i+1)%(num\_doors)

  return i

def monty\_hall\_game(switch, num\_tests):

  win\_switch\_cnt = 0

  win\_no\_switch\_cnt = 0

  lose\_switch\_cnt = 0

  lose\_no\_switch\_cnt = 0

  doors = [0,1,2]

  num\_doors = len(doors)

  for i in range(0,num\_tests):

    door\_with\_prize = random.randint(0, num\_doors-1)

    host = door\_with\_prize

    player\_choice = random.randint(0, num\_doors-1)

    original\_player\_choice = player\_choice

    shown\_door = get\_non\_prize\_door(host, num\_doors, player\_choice)

    if switch == True:

      player\_choice = switch\_function(shown\_door,num\_doors, player\_choice)

    if player\_choice == host and switch == False:

      print('Player Wins (No switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice ,', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

      win\_no\_switch\_cnt = win\_no\_switch\_cnt + 1

    elif player\_choice == host and switch == True:

      print('Player Wins (switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

      win\_switch\_cnt = win\_switch\_cnt +1

    elif player\_choice != host and switch == False:

      print('Player Lost (No switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

      lose\_no\_switch\_cnt = lose\_no\_switch\_cnt + 1

    elif player\_choice != host and switch == True:

      print('Player Lost (switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

      lose\_switch\_cnt = lose\_switch\_cnt + 1

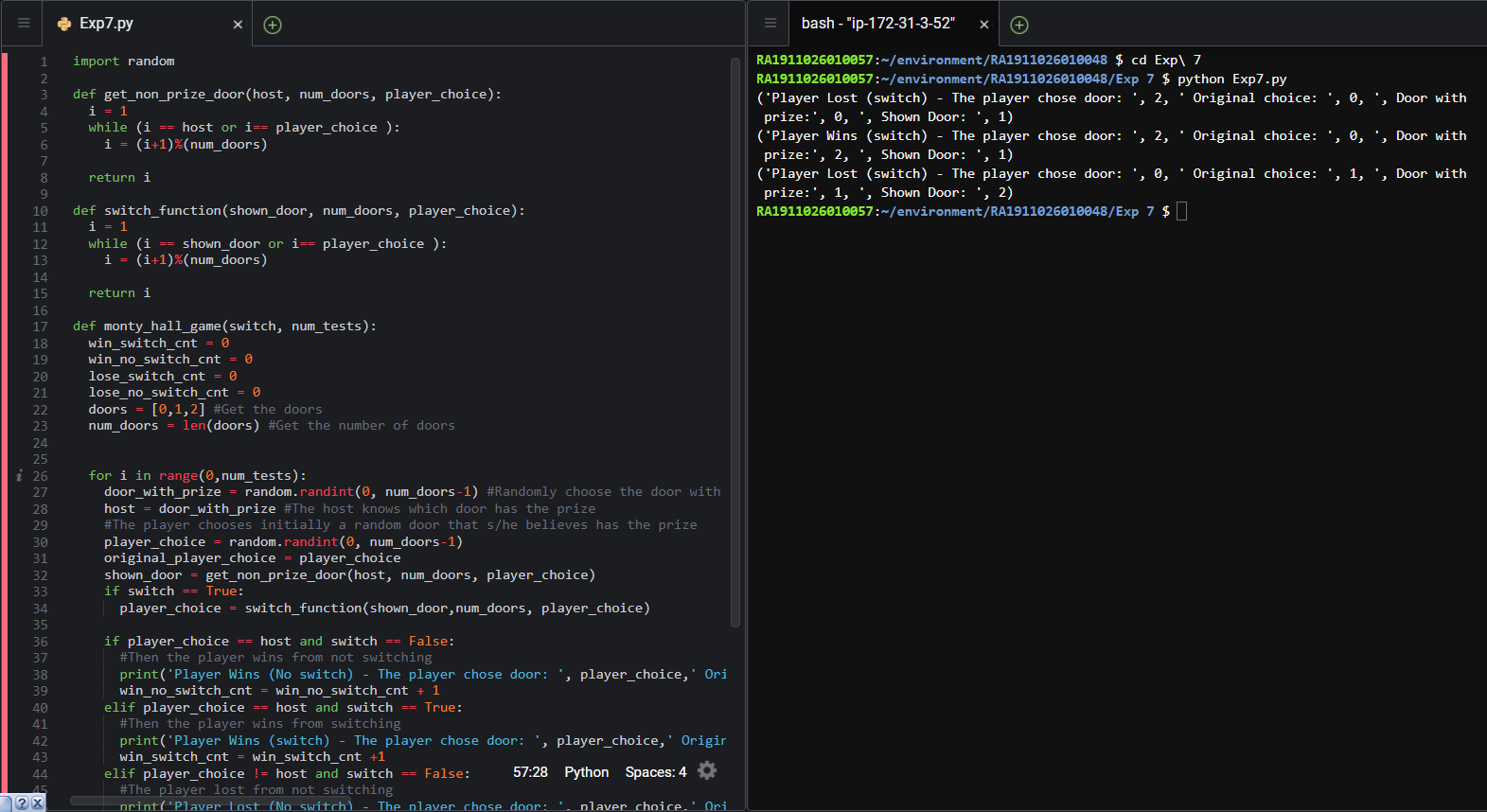
    else:

      print('SOMETHING IS WRONG')

  return win\_no\_switch\_cnt,win\_switch\_cnt,lose\_no\_switch\_cnt,lose\_switch\_cnt, num\_tests

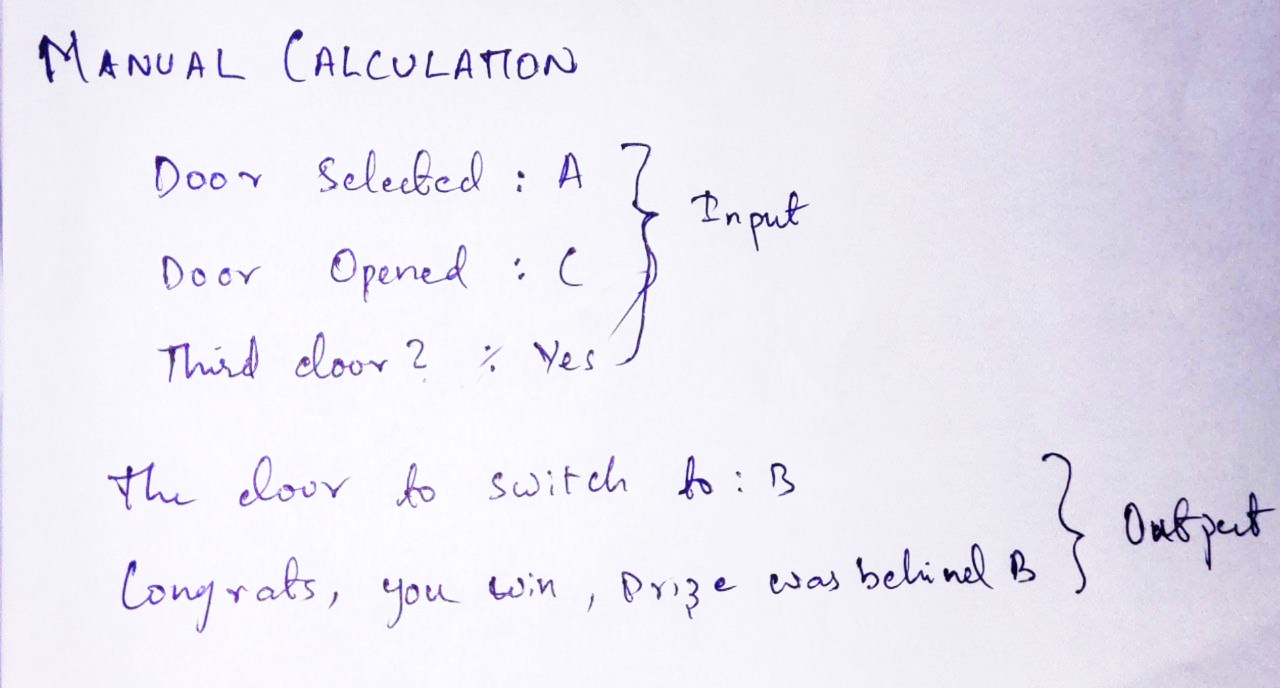
x = monty\_hall\_game(True,3)

**Output:**



**Observations:**

By Manual Calculation,



**Inference:**

The Monty Hall Problem was implemented successfully through a game of choosing doors and winning a car in the end.

**Result:**

Monty Hall problem is implemented successfully.