

GNG1106 Lab 3

The Monty Hall Problem

Background

The Monty Hall Problem is a famous probability puzzle, based on an old television show, in which the host, Monty Hall, plays a game with a player. It goes like this: In each game, three doors are presented to the player, where behind one of the three doors, there is a prize and behind the other two doors there is nothing. Monty Hall then asks the player to pick a door; the game rule is that the player will be given the prize if he picks the door behind which there is the prize. After the player picks a door, say Door 1, Monty Hall will take a peek at what is behind the other two doors and open one of them. He would say, "Look, there is no prize behind this door!". Note that since only one door has a prize, it is always possible for Monty Hall to pick a door that the player did not pick and that has no prize. Suppose that the door he opens is Door 2. Monty Hall then goes on and says, "I am going to make you a deal: I will allow you to change your decision by switching your pick to the other door (in this example, to Door 3). Do you want to do that?" It is up to the player to decide if he wants to switch to Door 3 or to stay with Door 1. After the player makes his switch-or-not-switch decision, Monty Hall opens the door of the player's final choice, and the player will get a prize if an only if there is the prize behind the door of his choice.

Historically there were intense debates on whether switching would improve the chance of winning, which made the TV show very popular. In this lab, you are to develop a program that simulate the Monte Hall game and investigate this problem by yourself.

Description

The program you are to write include two modes: game mode and research mode. Briefly, in the game mode, the program will act as Monty Hall and plays the game with the program user; in the research mode, the program allows the user to study what is the probability of winning the prize for the always-switch strategy and for the never-switch strategy.

When the program runs, it first prints a menu and prompts user to select an option:

Enter one of the following options:

1. Game Mode
2. Research Mode
3. Exit

Only when the user chooses to exit will the program exit. If the user chooses the game mode and finishes playing a game, the program comes back to this menu and prompts the user to enter an option again. Similarly, if the user chooses the research mode and finishes the research on either one of the two strategies, the program also comes back to this menu and prompts the user again.

Game Mode

In the game mode, the program acts as Monty Hall and simulates ONE game play with the user.

The three doors will be denoted by Door 1, Door 2 and Door 3. The program starts by selecting a door from the three at (uniformly) random and “putting” the prize behind that door. It then asks the user to pick a door. After the user makes his pick, the program, based on what the user picks, “opens” a door that user did not pick and that has no prize behind it, and then asks the user if he wants to switch. After the user makes his switch-or-not-switch decision, the program reveals where is the prize and announces if the user has won.

In addition to this main task, the total number of games that the user has played in the game mode and the number of times he wins should also be counted and printed out after each game. Note that even if a user leaves the game mode to access the research mode, his game play statistics in the game mode (namely, total number of games played and the number of winnings in the game mode so far) should still be kept, so that when the user comes back to the game mode and play the game again, the counting can be continued.

Research Mode

In the research mode. the program first prompts the user to select from one of the two playing strategies, the *always-switch* strategy and the *never-switch* strategy. It also asks the user to enter the number N of game plays that the user would like to simulate for that strategy. Then for the selected strategy, the program simulates N game plays using that strategy.

In each game play, the program again acts as Monty Hall except that

1. no message is printed;
2. the program always sets the player’s initial pick to Door 1;
3. if the “never-switch” strategy is simulated, the program sets the player’s switch-or-not-switch choice always to “not switch”; if the “always-switch” strategy is simulated, program sets the player’s switch-or-not-switch choice always to “switch”.

That is, in the research mode, the actions of the player are also simulated by the program and there is no need for the user to enter anything in a game play.

After the simulation of N game plays, the program prints the percentage of games in which the user wins (which approximates the probability of winning, for large N).

Requirements

In addition to the regular requirements for labs, this lab requires you to modularize your code in terms of functions. In particular, noting that the game mode and the research mode have significant commonality, you should extract the common parts into function(s) and call them in both modes of the program.

In Step 3 of the lab where you need to develop a flow chart (or algorithm/pseudo code), it is better to split the flow chart into several flow charts hierarchically, namely, you may want to develop a high-level flow chart for the overall program and two flow charts each for a program mode. In addition, for each function you are to create, you should also develop a flow chart.