HW03 Problem Set

CS-3020

Assignment
Instructor Assigned: Exercise 8.0
Deitel and Deitel Exercise 8.18

HW03-1: (Instructor Assigned Exercise 8.0)

Write a simulator in which the computer acts as the 1970's "Let's Make a Deal" game show host Monty Hall. The rules are simple: There are three doors behind one of which is a new car and behind the other two are new goats. The contestant is asked to pick a door. At that point Monty (who knows where the car is) opens one of the other doors to reveal a goat. The player is then asked to stand or switch, meaning that they can stay with their original choice, or they can switch and choose the remaining unopened door. Once they make that decision, they receive whatever prize is behind their door.

The question to be explored is whether the person (assuming they want a car more than they want a goat) will maximum their odds of winning if they stick with their first choice, switch to the other door, or whether it doesn't matter.

Your program should take take command line arguments specifying a strategy and a number of games to play. The strategy is as follows:

0 – Interactive Play: Monty will ask the user to input their choices in real time.

1 – Always Switch: The player always chooses to switch doors.

2 – Always Stay: The player always chooses to stick with their original choice.

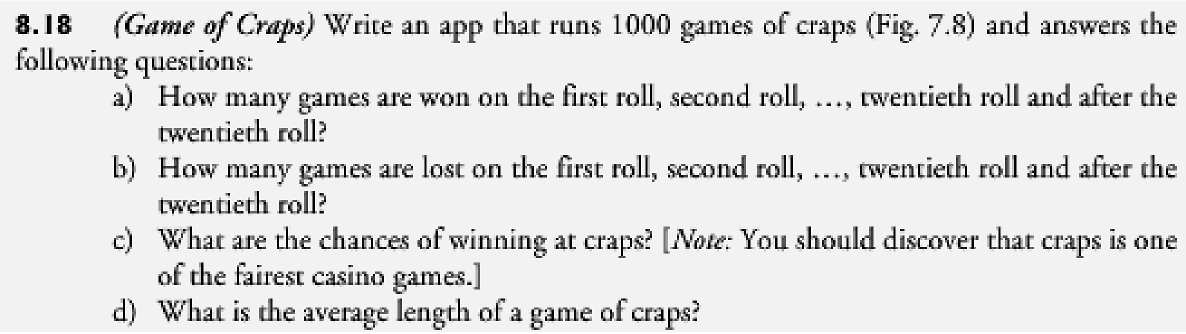
3 – Random: The player randomly chooses whether to stay or switch.

4 – Cycle: The game should play the specified number of games using each strategy in turn.

If no command line arguments are passed, then Interactive Play should be the default. If the first command line argument indicates Interactive Play, then the number of games, even if entered, should be ignored (in Interactive Play, Monty should ask the player if they want to play again). For the automated simulations, if a number of games is not given then the default should be one million.

The automated simulations should print to the console the win/loss percentage for the chosen strategy. If the Cycle strategy is chosen (the ultimate test) then the output should display the results for each strategy and, based on those, indicated which is the optimal strategy.

__HW03-2__: (Deitel & Deitel Exercise 8.18)



8.18 *(Game of Craps)* Write an app that runs 1000 games of craps (Fig. 7.8) and answers the following questions:

a) How many games are won on the first roll, second roll, …, twentieth roll and after the twentieth roll?

b) How many games are lost on the first roll, second roll, …, twentieth roll and after the twentieth roll?

c) What are the chances of winning at craps? [*Note:* You should discover that craps is one of the fairest casino games.]

d) What is the average length of a game of craps?

Supplemental Information

Be sure to review and adhere to the documentation and submission guidelines (see course website).

Hints and Suggestions

Exercise 8.0: Develop this app incrementally starting with Strategy 0 and working up. Also, keep in mind that your app does not need to take command line arguments or actually interact with the player from the get-go. First write an app that just plays the game against a hardcoded player that picks a specific door all the time and that always makes the same stay/switch decision. Get the basic game logic working first. Then get the interaction. Then get the automatic simulation for each single strategy working. Then get the combined simulation working. And then worry about processing command line arguments.

Grading Rubric

Each problem is worth 10 pts (score will be recorded as a percentage of that amount)

10% Properly submitted
10% Properly named
20% Adequate comments
10% Runs
20% Produces correct output
30% Effort evidenced by the submitted work