San José State University
Department of Computer Engineering

CMPE 180-92

Data Structures and Algorithms in C++

Fall 2016

Instructor: Ron Mak

Assignment #2B

Assigned:	Sunday, September 4
Due:	Friday, September 9 at 11:59 PM
URL:	http://codecheck.it/codecheck/files/16090406235c86ndhqrnrr23kaqtbawbkwy
Canvas:	Assignment 2.b. Monty Hall Puzzle
Points:	100

Monty Hall Puzzle

This assignment will give you practice decomposing a problem into smaller subproblems whose solutions you implement with functions. Submit a C++ program that simulates the Monty Hall Puzzle to demonstrate empirically whether it's better to stay with an original door choice or to switch doors (or it doesn't matter).

Go to the above URL for a partially written program **MontyHall.cpp** in CodeCheck's edit box. Complete the program and press the "Submit" button. You can type into CodeCheck directly, or you can edit and test your program outside of CodeCheck, such as in Eclipse or NetBeans, and then cut and paste into CodeCheck.

The list of function declarations is from a suggested solution. You are not obligated to use these declarations, but your program must be decomposed into functions.

You can submit as many times as necessary, and the number of submissions will not affect your score. When you're satisfied with your program, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution.

Submit the signed zip file into Canvas: Assignment 2.b. Monty Hall Puzzle.

Choose the right door and win the new car!

Monty Hall was a popular game show host on American television. In one segment of his show, you are a contestant from the studio audience. He shows you three doors, Door #1, Door #2, and Door #3. Hidden behind one door is a brand new car, and behind each of the other two doors is a goat.

Monty asks you to pick a door. Of course, you want to pick the right door and win the car.

Monty knows which door hides the car. After you've picked a door, he opens one of the other doors and shows you a goat.

Monty then offers you a chance to stay with the door you originally picked, or you can switch your choice to the remaining third door. Would staying or switching be better?

The simulation

Write a C++ program to simulate the above scenario. The program should:

1. Randomly pick a door to hide the car.
2. Acting as you, randomly pick a door as your first door choice.
3. Acting as Monty, open a door to reveal a goat. Since Monty (i.e., the program) knows which door hides the car:
    a. If your first door choice was correct, the other two doors each hides a goat. Monty randomly chooses one of the two doors to open.
    b. If your first door choice was incorrect (your door hides a goat), Monty opens the other door that hides a goat.
4. This leaves a third door to which you can switch as your second door choice. The program should keep track of whether you win by staying with your first door choice or if you win by switching to your second door choice.

Run the simulation 100 times. How many times do you win by staying with your first door choice vs. switching to a second door choice? Does it make a difference? What is the ratio of second door choice wins to first door choice wins?

Random number generation

As described in the above scenario, each simulation involves up to three random numbers. Use the predefined **`srand`** function to seed the random number generator at the start, then subsequently use the predefined **`rand`** function to generate the next random number. The generated pseudo-random numbers will be nonnegative integers.

Since the random numbers represent door numbers, you will need to modify the generated random numbers to be either 1, 2, or 3.

`srand`: http://www.cplusplus.com/reference/cstdlib/srand/
`rand`: http://www.cplusplus.com/reference/cstdlib/rand/

Sample output

Due to the random nature of the output, CodeCheck will not compare your program's output to the master. Here's sample output with 10 simulations (your program should do 100 simulations):

```
    #     Car   First  Opened  Second    Win     Win
         here  choice    door  choice  first  second

    1       3       1       2       3            yes
    2       2       1       3       2            yes
    3       2       2       1       3    yes
    4       2       2       3       1    yes
    5       1       1       3       2    yes
    6       3       2       1       3            yes
    7       2       2       1       3    yes
    8       1       3       2       1            yes
    9       2       3       1       2            yes
   10       1       3       2       1            yes

    4 wins if stay with the first choice
    6 wins if switch to the second choice


 Win ratio of switch over stay: 1.5
```

For many, the results are counterintuitive. A statistician can explain the results using conditional probabilities. See https://en.wikipedia.org/wiki/Monty_Hall_problem