Programming Languages, Fall 2023

Project #1 Due 9/29/23 (email file to me)

Description: Write a C program to estimate the probability of all 100 prisoners escaping in the 100 Prisoner Problem. In this problem, 100 prisoners, numbered 1 to 100 are imprisoned. A room contains 100 closed numbered boxes which contain the numbers 1 to 100 in some order. Each box contains just one of the numbers. Each prisoner is given the opportunity of finding their number by opening a maximum of 50 boxes. No prisoner sees what any other prisoner does, but the prisoners are able to discuss strategy ahead of time. If all 100 prisoners find their number, then all 100 prisoners are set free. If even one fails to find their number then all 100 prisoners are executed. What is the optimal strategy and what is the resulting probability that the prisoners all escape? See <https://en.wikipedia.org/wiki/100_prisoners_problem> for what the solution is.

Your job is to write a C program which implements the solution and confirms the resulting probability of success. You will represent the room with the boxes as an array of length 100 consisting of the numbers 1 to 100. Your code should contain functions for the following:

1. Initialize the room with a random permutation of 1-100
2. Given a prisoner number and the room, determines if the prisoner finds their number in at most 50 steps
3. Given a room, determines if all prisoners find their numbers

Given these functions, your code is to run the last function on a large number (e.g. 100,000) of randomly chosen rooms and estimate the probability that all prisoners find their number. Print this probability. Email a file named

Extra credit includes

1. Find a clever way to implement the third function above, one that doesn’t simply call the second function in a naïve loop. Hint: if a prisoner finds their number in a chain of 20 numbers, all of the other prisoners on the chain will find their number.
2. Implement your own random number generator, one which is better then the standard library rand(). Alternatively, find a way to use the Mersenne Twistor using code that you include with your project.
3. Investigate what happens as the number of prisoners tends to infinity. For example, compile a table of values for 100, 200, 300, …, 2000
4. Anything else which you can think of which is in C and clearly related to the project.