Code Challenge: Mastermind

Résumé:
This project is an AI program to get you to do automated data analysis and AI decision making.

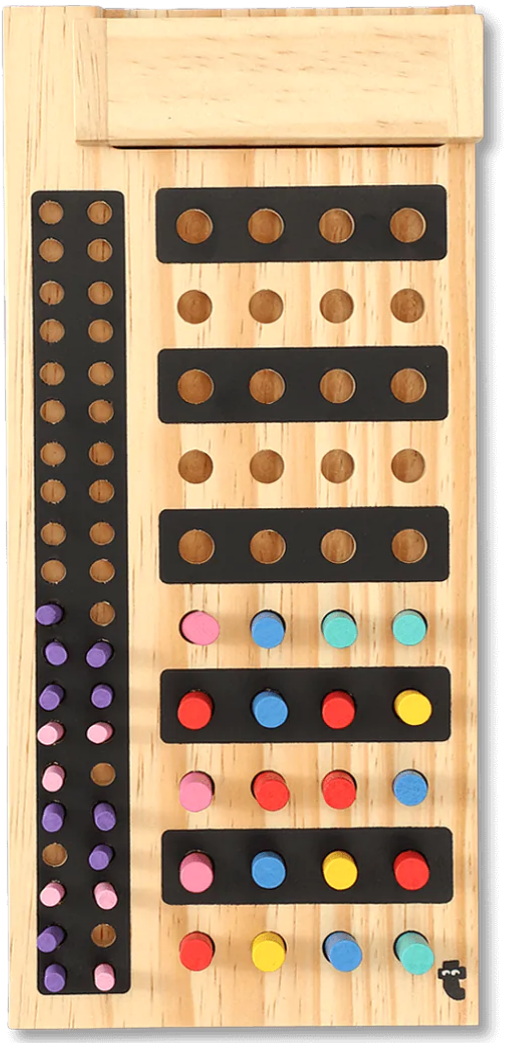
Table des matières

I Introduction

Mastermind is a game where one player (**the setter**) sets a sequence (code) of 4 colors - with possible repetition - out 5 to 10 colors.

The second player (**the breaker**) must then try to find the correct code by setting their own sequence of 4 colors on the revealed side of the board (with 6 to 12 attempts).
The setter responds to each breaking attempt with two numbers. The numbers represent the number of correctly placed correct colors in the attempt (in purple pins) and a number of incorrectly placed correct colors (pink pins).

The breaker must find the correct code with the least number of attempts.
The game is repeated to find which of the two players is the better code breaker.



II Objective

The challenger must create a program that will receive the number of letters that can appear in the game, the number of positions in the code, and the number of attempts the program can use to find the correct code that the user is thinking about.
The code is made of X number of letters with the possibility of repetition.

The program will then repeatedly guess the code. The user will then be prompted to input how many correct letters appear in the proposed sequence in their correct position and how many appear in the wrong position.

After the user inputs the number X in response to the “correct letter in correct position” or after the program uses its last attempt, the program will terminate.

III Partie obligatoire

Nom du programme	mind_breaker
Fichiers de rendu	any files in any language as long as they are written by the challenger.
Arguments	(int) The number of possible letters appearing in the code, (int) the number of positions in the code, (int) the number of attempts possible.
Description	Create a program that will guess the correct code using the possible letters specified by the user and analyze the responses of the user until finding the code or running out of attempts.

The program follows the following specifications:

- The program receives 3 integers, the first **(X)** determines the letters appearing in the game (the letters start from a). The second is how many positions are there in the code (how many characters the code is made of). The third is how many times the program will guess and receive the response from the user.
- The program will put out X letters.
- The program will prompt the user to input an integer for how many correct letters in their correct position in the number. The program will prompt the user to input another integer for how many correct letters appear in the wrong position.
- The process repeats until the user inputs X for the number of letters in the correct position, or until the program uses all its attempts.

IV Evaluation and rewards

While all entries need to fulfill the obligatory conditions, the winners of the challenge will be the ones to score the top in one of the four following criteria:

1. **Fast brain**: your program must find the solution in the shortest amount of time when there are 4 positions and 10 available letters.
2. **Big Brain**: Your program must solve with the least amount of attempts on average for 20 positions and 10 letters.
3. **Sharp brain**: your program must display a graphical representation with the highest number of conclusions that drive its decision making process at the end of the game.
4. **Professor**: Impress us when you have to solve 20 positions with 20 colors.

Challenge evaluation	4 ⊙	20 ⊙
10 letters	Fast brain ⏲	Big brain ↺
20 letters	Sharp brain 📊	Professor 🧠

A custom banner will be offered to the winner of each category. Titles may be offered depending on the level of participation and approval of the bocal.

An extra reward is offered to anyone willing to create a tester for the event (or offers assistance in creating it.

V Examples

Example 1: The user is thinking 3153. They tell the program that there are up to 5 letters in their number, and that the program has 10 attempts to try and find the right number.

```
$>./mind_breaker 5 4 10
Attempt 1
1212
How many correct letters are in the correct position?
>0
How many correct letters are in the incorrect position?
>1
Attempt 2
3455
How many correct letters are in the correct position?
>1
How many correct letters are in the incorrect position?
>1
Attempt 3
3314
How many correct letters are in the correct position?
>1
How many correct letters are in the incorrect position?
>2
Attempt 4
5555
How many correct letters are in the correct position?
>1
How many correct letters are in the incorrect position?
>0
Attempt 5
3153
How many correct letters are in the correct position?
>4
I win!
```

The user responds to Attempt 1 with 0 letters in the correct position since the 1 letter is not in the second position. And they respond 1 for only 1 letter (the second one doesn't count since there is only one 1 in the number).

The user responds to Attempt 3 with 1 letter representing one of two 3s in its correct position. And 2 for the other three and for the 1.

Example 2: The user is thinking 4444. They tell the program that there are up to 5 letters in their number, and that the program has 10 attempts to try and find the right number.

```
$>./mind_breaker 5 5 10
Attempt 1
12121
How many correct letters are in the correct position?
>0
How many correct letters are in the incorrect position?
>0
Attempt 2
34555
How many correct letters are in the correct position?
>1
How many correct letters are in the incorrect position?
>0
Attempt 3
33333
How many correct letters are in the correct position?
>0
How many correct letters are in the incorrect position?
>0
Attempt 4
44444
How many correct letters are in the correct position?
>5
I win!
```