Mastermind

PYTHON IMPLEMENTATION

Jan 6, 2022

Revision History

Version No	Date	Description
1.0	1/6/2022	Initial Creation

Introduction

Mastermind is an old code-breaking game played by two players. The game goes back to the 19th century and it resembles an earlier pencil and paper game called Bulls and Cows that may date back a century. The modern game with pegs was invented in 1970 by Mordecai Meirowitz, an Israeli postmaster and telecommunications expert. Usually, the mastermind game is color-coded but for this program, instead of using different colors, we represent it with different numbers.

Rules of the Game

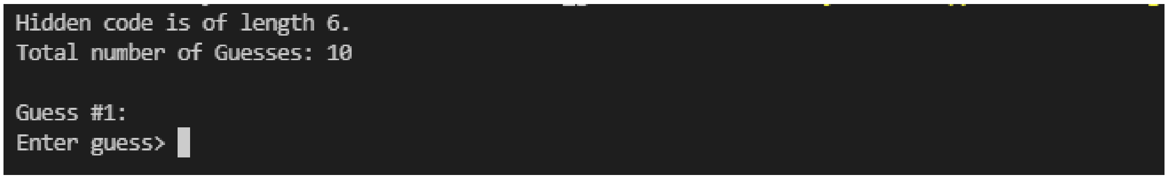
The game implementation is a one player game with the player as the codebreaker against a computer (CPU) as codemaker. The rules should be as follows:

- The CPU randomly generates a pattern of numbers with a length of 4 to 8 digits. NOTE: the pattern may contain duplicate numbers (e.g. 1014)
- The player then attempts to guess the pattern with the corresponding length of the pattern provided before the start of the game.
- For every guess of the player, the CPU will return a response hinting the number of digits in a correct position and the digits that are in the pattern but not in the correct position.
    - Red indicator - indicates the number of digits in a correct position
    - White indicator - indicates the digits that are in the pattern but not in the correct position
- Aside from pattern guesses from the player, the player may also use lifeline to help on identifying the pattern. In using lifeline, the player also risks reducing the number of guesses that the player can use. Below are the lifelines available:

  - The player can ask the program to reveal a number in the generated code.
  - The player can ask the program to reveal the correct color in one position.
- The game ends when the player correctly identifies the pattern, wins the game and is crowned Mastermind! If not, then the game will continue to ask for guesses from the player until the pattern is correctly identified or the maximum number of guesses is already used up.
- The real game, however, has proven aesthetics since the numbers are represented by color-coded buttons.

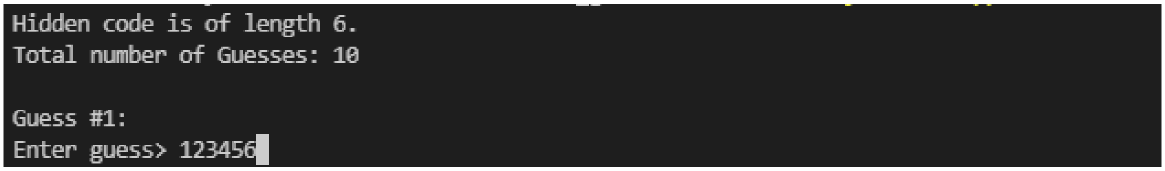
How to Play with the Program

1. At the start of the game, the program or the CPU will display the length of the hidden pattern and the total number of guesses the player can use.



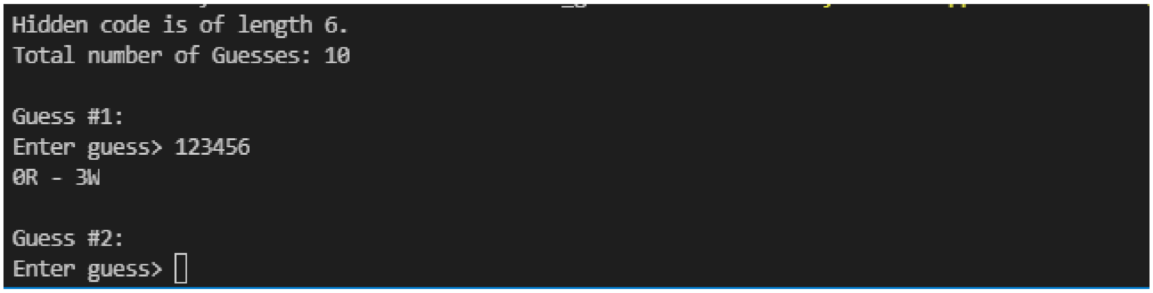
```
Hidden code is of length 6.
Total number of Guesses: 10

Guess #1:
Enter guess>
```

2. The player will then input the guess or the possible pattern on the input field as shown below:



```
Hidden code is of length 6.
Total number of Guesses: 10

Guess #1:
Enter guess> 123456
```

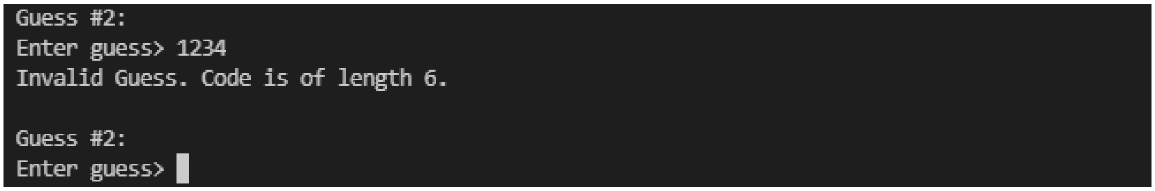
3. Based on the input, the computer will then feedback the result of the guess hinting at the number of digits in a correct position and the digits that are in the pattern but not in the correct position with the indicators below:
    a. Red indicator (*R) - indicates the number of digits in a correct position
    b. White indicator(*W) - indicates the digits that are in the pattern but not in the correct position



```
Hidden code is of length 6.
Total number of Guesses: 10

Guess #1:
Enter guess> 123456
0R - 3W

Guess #2:
Enter guess>
```

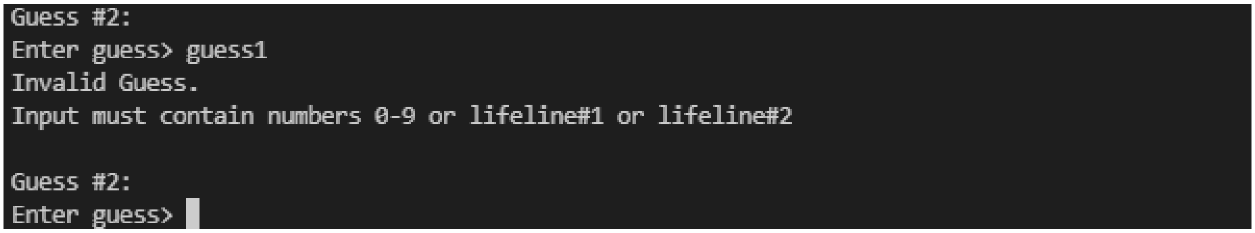
In this example, **0R** indicates that there are no correct numbers in correct position and **1W** indicates that there is a single number in the guess that is also in the pattern given by the CPU

4. If the player inputs an invalid length of digits, it will print an error and will ask the player to input a guess again with the correct length.



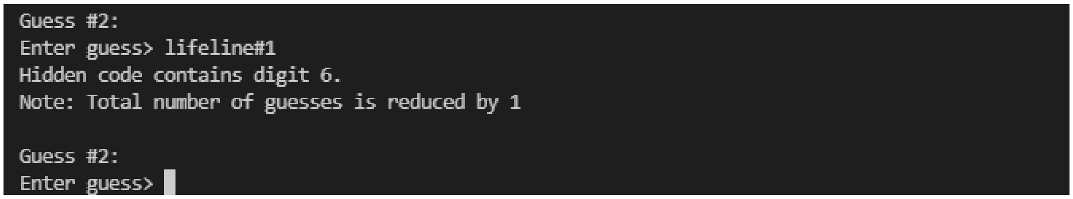
```
Guess #2:
Enter guess> 1234
Invalid Guess. Code is of length 6.

Guess #2:
Enter guess> 
```

5. If the player inputs and invalid characters aside from numbers, it will print an error and will ask the player to input a guess again with a correct input containing only numbers.



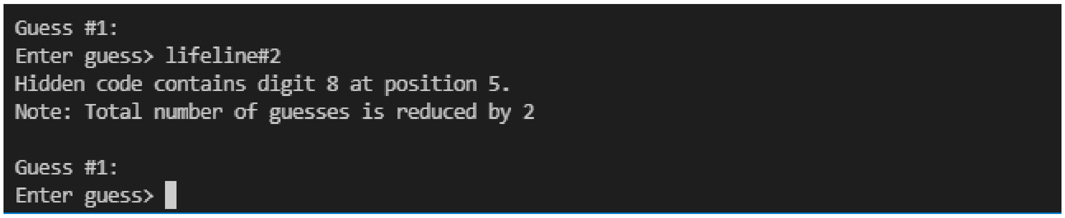
```
Guess #2:
Enter guess> guess1
Invalid Guess.
Input must contain numbers 0-9 or lifeline#1 or lifeline#2

Guess #2:
Enter guess> 
```

6. The player may also use lifeline to help on breaking the pattern given by the CPU. These lifelines may ONLY BE USED ONCE during the whole duration of the game. If another lifeline is requested by the player, an error message will be displayed and will ask again for the player's input to provide guesses only. Lifelines can be accessed by entering the following in the input field:
    a. lifeline#1 : The player can ask the program to reveal a number in the generated code. As a consequence, the available number of guesses for the player will be decreased by ONE.



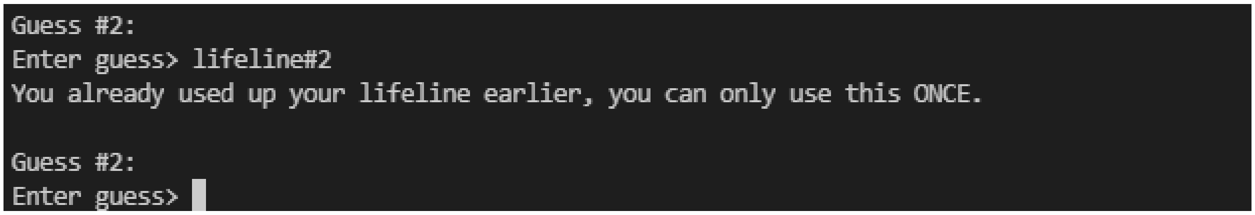
```
Guess #2:
Enter guess> lifeline#1
Hidden code contains digit 6.
Note: Total number of guesses is reduced by 1

Guess #2:
Enter guess> 
```

    b. lifeline#2 : The player can ask the program to reveal a correct number and its position. As a consequence, the available number of guesses for the player will be decreased by TWO.



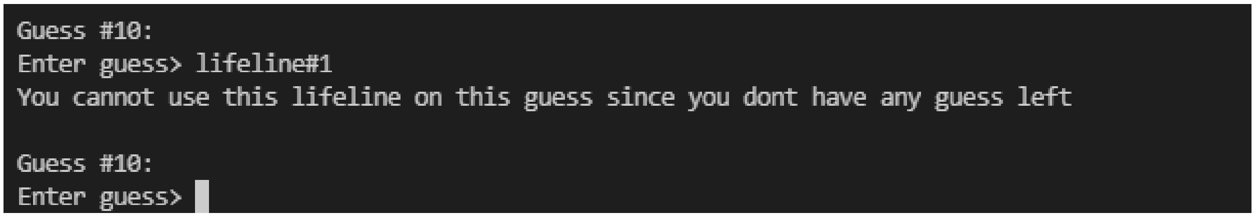
```
Guess #1:
Enter guess> lifeline#2
Hidden code contains digit 8 at position 5.
Note: Total number of guesses is reduced by 2

Guess #1:
Enter guess>
```

Given the following lifelines, the player is NOT allowed to use a lifeline when the lifeline will exhaust the number of available guesses. For example: The player is not allowed to use the lifeline#2 during guess #9 and #10 since it will already exhaust the number of guesses.

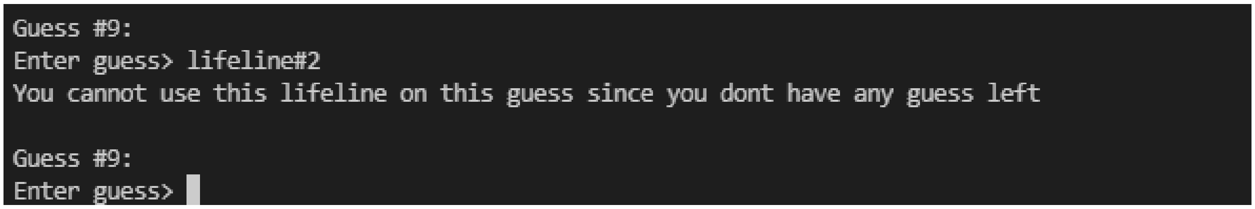


```
Guess #2:
Enter guess> lifeline#2
You already used up your lifeline earlier, you can only use this ONCE.

Guess #2:
Enter guess>
```

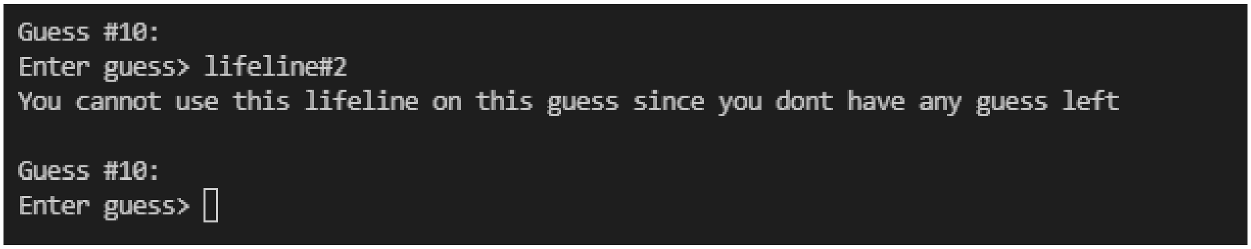
An error message “You already used up your lifeline earlier, you can only use this ONCE.” will be displayed if the user requested for a lifeline more than once.



```
Guess #10:
Enter guess> lifeline#1
You cannot use this lifeline on this guess since you dont have any guess left

Guess #10:
Enter guess>
```



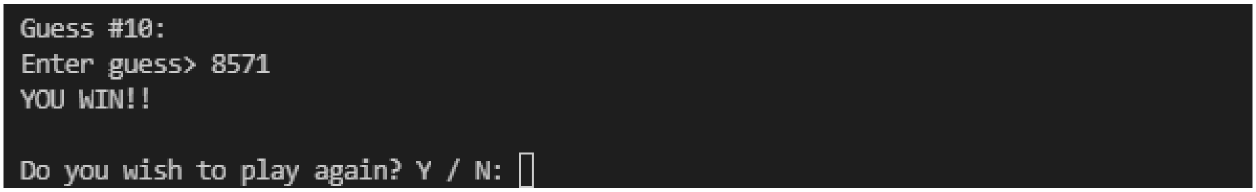
```
Guess #9:
Enter guess> lifeline#2
You cannot use this lifeline on this guess since you dont have any guess left

Guess #9:
Enter guess>
```



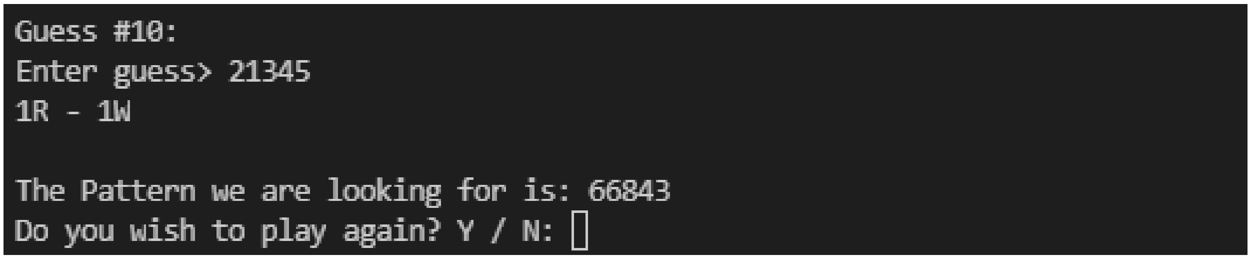
```
Guess #10:
Enter guess> lifeline#2
You cannot use this lifeline on this guess since you dont have any guess left

Guess #10:
Enter guess>
```

An error message “You cannot use this lifeline on this guess since you don’t have any guess left” will be displayed if the user requested for a lifeline even if it will exhaust the remaining guess available.

7. The game ends when the player successfully guessed the pattern before the number of guesses runs out and wins the game or if the number of guesses runs out before the player successfully guessed the pattern. If the pattern is not successfully guessed, the CPU will display the correct pattern generated.

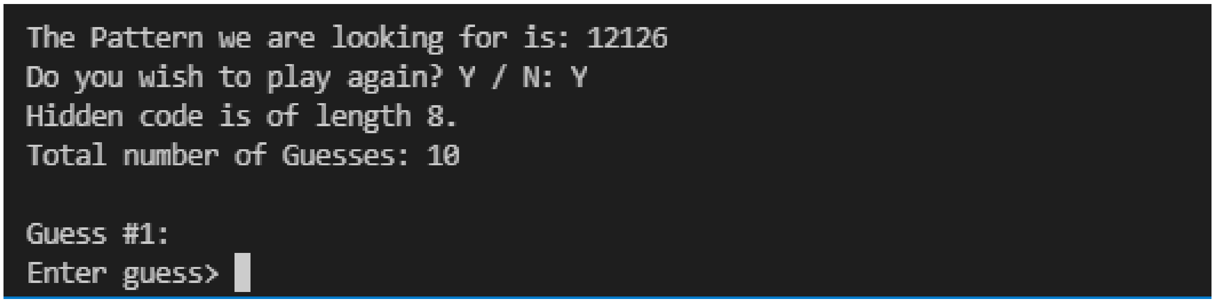


```
Guess #10:
Enter guess> 8571
YOU WIN!!

Do you wish to play again? Y / N: 
```



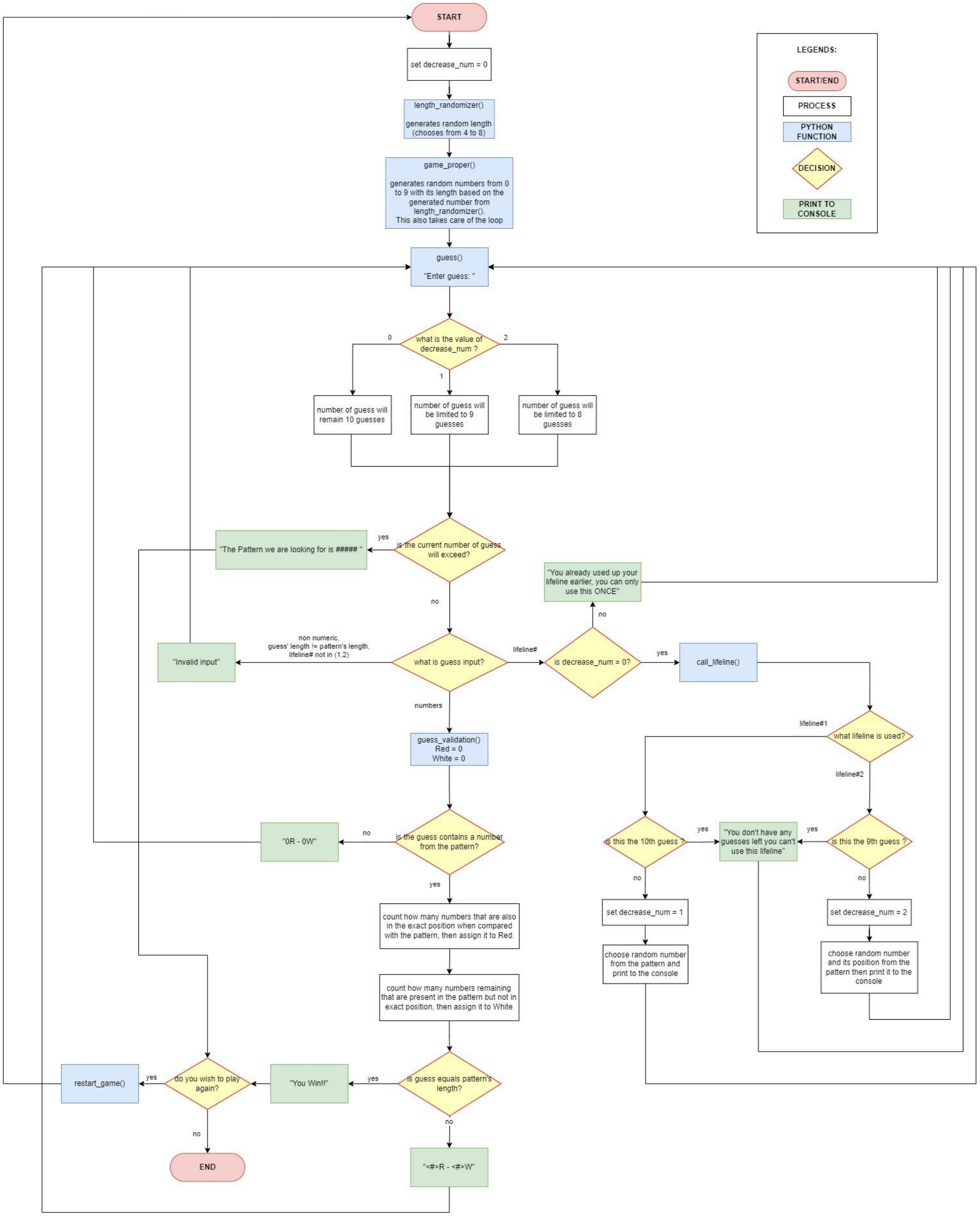
```
Guess #10:
Enter guess> 21345
1R - 1W

The Pattern we are looking for is: 66843
Do you wish to play again? Y / N: 
```

8. At the end of the game, the CPU will ask the player if the player wants to play again with the game. If the player inputs ‘Y’ for Yes, it will generate another pattern and will then resume the game. If not, then the program ends and exits the game.



```
The Pattern we are looking for is: 12126
Do you wish to play again? Y / N: Y
Hidden code is of length 8.
Total number of Guesses: 10

Guess #1:
Enter guess> 
```

Implementation


START
set decrease_num = 0
length_randomizer()
generates random length (chooses from 4 to 8)
game_proper()
generates random numbers from 0 to 9 with its length based on the generated number from length_randomizer().
This also takes care of the loop
guess()
"Enter guess: "
what is the value of decrease_num ?
0
1
2
number of guess will remain 10 guesses
number of guess will be limited to 9 guesses
number of guess will be limited to 8 guesses
is the current number of guess will exceed?
yes
"The Pattern we are looking for is ##### "
no
what is guess input?
non numeric, guess' length != pattern's length, lifeline# not in (1,2)
"Invalid input"
lifeline#
is decrease_num = 0?
no
"You already used up your lifeline earlier, you can only use this ONCE"
yes
call_lifeline()
what lifeline is used?
lifeline#1
lifeline#2
is this the 10th guess ?
yes
"You don't have any guesses left you can't use this lifeline"
yes
is this the 9th guess ?
no
set decrease_num = 1
choose random number from the pattern and print to the console
no
set decrease_num = 2
choose random number and its position from the pattern then print it to the console
numbers
guess_validation()
Red = 0
White = 0
is the guess contains a number from the pattern?
no
"0R - 0W"
yes
count how many numbers that are also in the exact position when compared with the pattern, then assign it to Red.
count how many numbers remaining that are present in the pattern but not in exact position, then assign it to White
is guess equals pattern's length?
yes
"You Win!!"
do you wish to play again?
yes
restart_game()
no
END
no
"<#>R - <#>W"
LEGENDS:
START/END
PROCESS
PYTHON FUNCTION
DECISION
PRINT TO CONSOLE

References

- Mastermind Game - https://en.wikipedia.org/wiki/Mastermind_(board_game)
- Python random - https://www.w3schools.com/python/module_random.asp