Higher Lower Game

# Scenario

You have a younger family member who loves the ‘higher / lower’ game so you decide to make a game that they can play whenever they want.

# No Frills Game

Your game should…

* Generate a ‘secret’ number between 1 and 100 and then ask the user to guess the number.
* Tell the user if their guess is ‘too high’ or ‘too low’ until either the number has been guessed or the user has used tried to guess more than 10 times.
* If the user correctly guesses the number, the game should congratulate them.
* If they don’t guess the number in 10 goes, they should be told that they have lost the game and the mystery number should be revealed.
* At the end of each game, the user should be asked if they would like to play again
* Ideally the game should be set up so that users can’t guess the same \*wrong\* number twice.

|  |
| --- |
| Variations (recommended) Please consider adding in the following features. You should develop (and test) these as part of the problem decomposition.   * Allow the user to choose the lowest and highest number that will be used as the secret number * Either allow the user to choose the number of guesses or set a sensible limit based on the difference between the highest and lowest number. * Ask the user how many rounds they would like to play in a given game. Choose a sensible maximum for this and justify your decision * Keep track of the number of rounds played **and** the user’s scores * At the end of the given number of rounds, tell the user what their best, worst and average scores were. Ask them if they wish to play again |

## Task

1. Decompose the problem (write down the decomposition on the template supplied)
2. For each part of the problem, write (and test) each piece of code. Before you write a piece of code, you should generate a quick test plan so that you can confirm that the code works correctly. Place your test plan and testing evidence on the supplied template.
3. Combine your code into a fully working program
4. Test and debug your program to ensure that it works for expected, boundary and unexpected values
5. Ask a friend / parent to play your game. Watch them as they do this and make note of any changes that could be made to make the game easier to use
6. Make the changes identified in the previous step
7. Retest your game to ensure that it still works correctly