**NUMBER GUESSING GAME**

You will create a number guessing game where the user will guess a number in a range e.g. between 1 and 100 (inclusive). The program will generate a random number and ask the user to guess the number. Give them some help. If the guess needs to be bigger, tell them "It's bigger". If it needs to be smaller, say "It's smaller".

Your program will ask the user if they want to play on an easy mode or hard mode. On easy mode, they have a limit of 10 guesses, but on hard mode they will only have 4 guesses. If they guess it correctly within the limit, the program will say that they have won. If they don’t get it right within the limit, they lose.

Plan, code and comprehensively test your program. It is important that you show how you decomposed the outcome into smaller components. This may be communicated using a range of tools eg natural language (as long as it is unambiguous and precise), diagrams (including screen layouts), and pseudocode.

Throughout the development of your program code, ensure that you document your program with appropriate variable/module names and comments that describe code function and behaviour. Follow the common conventions of your programming language (e.g. naming conventions or rules for program layout).

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Guess - integer | Checking if guess is right or not | You were right / you were wrong |
| What mode - string or an int corresponding to the value> (e.g. 1 - easy, 2 - hard) | Is the guess lower or higher than the number | It’s bigger / it’s smaller |
|  | Do they have any guesses left | You lose - you ran out of guesses / you win - you guessed right |

**Algorithm**

1. Import random
2. Ask the user what mode they want to play in
   * If easy, set guesses to 10
   * If hard, set guesses to 4
3. Select a random number from 1 to 100 (inclusive)
4. Input - get the user to guess a number
   * If not an integer, try again
   * If not in range, try again
5. Check if guess == number
   * If it is, print that they were right and they won, and end program
   * If it isn’t, move on to next step
6. Guesses -= 1
7. If guesses != 0, carry on
   * Otherwise, print that they lost and end program
8. If guess < number
   * Print that the number is bigger
9. If guess > number
   * Print that the number is smaller
10. Return to step 4

**Variables:**

* EASY\_GUESS - constant - how many guesses are in easy mode
* HARD\_GUESS - constant - how many guesses are in hard mode
* mode - will decide whether the mode is easy or hard, could be something like “e” or “h” or 1 or 2
* guess\_num - how many guesses the user has - will be set to EASY\_GUESS or HARD\_GUESS (will decrease throughout program)
* num - the random number that the user has to guess (will be done through randint)
* guess - the guess that the user made, will be input

*I’m not sure yet which variables will be local and which will be global*

**Versions:**

1. Receiving and storing input, checking if it is the same as random number
2. Looping until user gets random number
3. Adding the “the number is lower, the number is higher” messages
4. Adding a guess counter
5. Adding easy and hard modes
6. Error handling

**Functions:**

Possible functions could be:

* guess\_input(msg) - forces int input within 1-100 range

**Testing Plan:**

|  |  |
| --- | --- |
| **Test** | **Result** |
| Guess Input - Achieved | |
| 50 | \*program carries on as normal\* |
| 75 | \*program carries on as normal\* |
| Guess Input - Merit | |
| 0 | Please enter a number between 1 and 100 |
| 1 | \*program carries on as normal\* |
| 2 | \*program carries on as normal\* |
| 99 | \*program carries on as normal\* |
| 100 | \*program carries on as normal\* |
| 101 | Please enter a number between 1 and 100 |
| Guess Input - Excellence | |
| -3 | Please enter a number between 1 and 100 |
| 4.5 | Please enter a valid integer |
| hello | Please enter a valid integer |
| 130 | Please enter a number between 1 and 100 |
| \*enter\* | Please enter a valid integer |
| Mode Input - Achieved | |
| e | \*starts easy mode\* |
| h | \*starts hard mode\* |
| Mode Input - Excellence | |
| E, e , (capitals, spaces, etc.) | \*starts easy mode\* |
| H, h , (capitals, spaces, etc.) | \*starts hard mode\* |
| easy | Please enter either E for easy mode or H for hard |
| \*enter\* | Please enter either E for easy mode or H for hard |
| 37 | Please enter either E for easy mode or H for hard |
| hello | Please enter either E for easy mode or H for hard |