

Pseudocode for Individual Programming Assignment: Password Guessing Game

Main Program:

Set `candidate_words` list

Display welcome message

Endless Loop

Prompt user for `difficulty` choice (e, m, or h)

If `difficulty` is 'e'

 Display selected difficulty (easy)

 Set `num_words` to 7

 Set `word_list` to a list of random words from `candidate_words` which is `num_words` long

 Set `guesses_remaining` to 5

 Break out of loop

However, if `difficulty` is 'm'

 Display selected difficulty (medium)

 Set `num_words` to 8

 Set `word_list` to a list of random words from `candidate_words` which is `num_words` long

 Set `guesses_remaining` to 4

 Break out of loop

Or if `difficulty` is 'h'

 Display selected difficulty (hard)

 Set `num_words` to 9

 Set `word_list` to a list of random words from `candidate_words` which is `num_words` long

 Set `guesses_remaining` to 3

 Break out of loop

Otherwise

 Display Invalid message (Loop prompts again)

Gameplay loop pseudocode on next page.

Set `won` to False

Set `password` to a random word in `word_list`

Set `guesses_amount` to the `guesses_remaining`

While `guesses_remaining` is greater than 0 and `won` is False

- Display a message stating the password is in the following list

- Display and repeat for each word from `word_list`

- Display the remaining guesses from `guesses_remaining`

- Obtain and validate the users `guess` by passing it through the `input_int` function

- Display the users `guess` as a string stored in `guessed_word`

- If the `guessed_word` is not the `password`

 - Display incorrect password message

 - Invoke `compare_words` and compare the `password` and `guessed_word` to return `letters_in_common`

 - Display `letters_in_common`

 - Reduce `guesses_remaining` by 1

- Otherwise

 - Set `won` to True

If `won` is True

- Display 'you win' message

- And, if `guesses_remaining` is the same as `guesses_amount`

 - Display 'lucky guess' message

Otherwise

- Display 'you lose' message

- Display `password`

Functions pseudocode on next page.

Functions:

Define 'compare_words' function (receives "word_1" and "word_2")

- Set letters to 0

- Repeat for the length of word_1

 - Check if the letters in word_1 and word_2 are the same for each position

 - If so, increase letters by 1

- Return letters

Define 'input_int' function (receives "prompt", "minValue = 1" and "maxValue = None")

- Endless Loop

 - Prompt user for their guess

 - Try convert guess into an integer and store as int_guess

 - Except a value error

 - Display 'invalid input' message for not being an integer

 - Continue through loop

 - If int_guess is > maxValue or int_guess is < minValue

 - Display 'invalid input' integer is outside of the range

 - Continue through loop

 - Return the int_guess