Lone Star
section 4

Advanced Functions quiz

1. (9.1) (Game: Odd or Even) Write a **function** that lets the user guess whether a randomly generated number is odd or even. The function randomly generates an integer between 0 and 9 (inclusive) and returns whether the user's guess is correct or incorrect. The argument for the function will be *guess* (the user's guess, either "odd" or "even"), if no argument is provided then the **default** guess should be even.
   Hint: Use the following lines of code to create the function.

   ```
   from random import randint
   value = randint(0,9) #picks a random integer between 0-9 inclusive
   ```

   Examples:

   - guess( ) → "Correct!" (if random value is even) or "Incorrect!" (if random value is odd)
   - guess("odd")→"Correct!" (if random value is odd) or "Incorrect!" (if random value is even)
   - guess("even") → "Correct!" (if random value is even) or "Incorrect!" (if random value is odd)

2. (9.2) Write a **function** named *is_two_digit_number* that returns a boolean value which determines if an integer is a two digit number. Write a second function named *report_two_digit_numbers* that takes a list of integers and returns a new list containing all the two digit numbers from the original list. Call the *is_two_digit_number* function as part of the *report_two_digit_numbers* function.

   Hint: a two digit number is one in the range $[-99, -10] \cup [10, 99]$.
   Examples:

   - report_two_digit_numbers([100,57,12,1]) → [57,12]
   - report_two_digit_numbers([121,36,-19,-6,0,21]) → [36,-19,21]
   - report_two_digit_numbers([100,7,8437]) → []

Dot Matrix
section 5

Advanced Functions quiz

1. (9.1) Write a **function** that takes two arguments, a list and a value. The function should return the indices of all occurrences of the *value* in the list, if no argument is provided then the **default** should be to find 0.

   Examples:

   - get_indices([1, 0, 5, 0, 7]) → [1, 3]
   - get_indices([1, 5, 5, 2, 7], 7) → [4]
   - get_indices([1, 5, 5, 2, 7]) → [ ]
   - get_indices([1, 5, 5, 2, 7], 5) → [1, 2]
   - get_indices([1, 5, 5, 2, 7], 8) → [ ]
   - get_indices(["a", "a", "b", "a", "b", "a"], "a") → [0, 1, 3, 5]

2. (9.2) Write a **function** named *is_vowel* that returns a boolean value which determines if an letter is a vowel. Write a second function named *report_vowels* that takes a string and returns a list containing all the vowels from the original string. Call the *is_vowel* function as part of the *report_vowels* function.

   Hint: In the English language, the letters a, e, i, o, and u are the vowels.
   Examples:

   - report_vowels("apple") → [a,e]
   - report_vowels("banana") → [a,a,a]
   - report_vowels("run time error") → [r,i,e,e,o]

Dark Helmet Advanced Functions quiz
section 4

1. (9.1) (Game: Odd or Even) Write a **function** that lets the user guess whether a randomly generated number is odd or even. The function randomly generates an integer between 0 and 9 (inclusive) and returns whether the user's guess is correct or incorrect. The argument for the function will be *guess* (the user's guess, either "odd" or "even"), if no argument is provided then the **default** guess should be even.
   Hint: Use the following lines of code to create the function.

   ```
   from random import randint
   value = randint(0,9) #picks a random integer between 0-9 inclusive
   ```

   Examples:

   - guess( ) → "Correct!" (if random value is even) or "Incorrect!" (if random value is odd)
   - guess("odd")→"Correct!" (if random value is odd) or "Incorrect!" (if random value is even)
   - guess("even") → "Correct!" (if random value is even) or "Incorrect!" (if random value is odd)

2. (9.2) Write a **function** named *is_even* that returns a boolean value which determines if an integer is even. Write a second function named *report_evens* that takes a list of integers and returns a new list containing all the even numbers from the original list. Call the *is_even* function as part of the *report_evens* function.

   - report_evens([4,3,12,16,8,9,25]) → [4,12,16,8]
   - report_evens([6,100,3,12,16,6,9,100]) → [6,100,12,16,6,100]
   - report_evens([3,99,7,13,25]) → []

President Skroob Advanced Functions quiz
section 1

1. (9.1) Write a **function** that returns the number of copies of the same number. The arguments for the function will be *num_1* (first number), *num_2* (second number), and *num_3* (third number), if no argument is provided then the **default** for all 3 values should be 0.

   Examples:

   - count_duplicates(2, 3, 2) → "There are 2 of the same number",
   - count_duplicates(4, 4, 4) → "There are 3 of the same number",
   - count_duplicates(1, 2, 3) → "Each number is unique"
   - count_duplicates(1) → "There are 2 of the same number"
   - count_duplicates(0) → "There are 3 of the same number"

2. (9.2) Write a **function** named *is_two_digit_number* that returns a boolean value which determines if an integer is a two digit number. Write a second function named *report_two_digit_numbers* that takes a list of integers and returns a new list containing all the two digit numbers from the original list. Call the *is_two_digit_number* function as part of the *report_two_digit_numbers* function.

   Hint: a two digit number is one in the range $[-99, -10] \cup [10, 99]$.
   Examples:

   - report_two_digit_numbers([100,57,12,1]) → [57,12]
   - report_two_digit_numbers([121,36,-19,-6,0,21]) → [36,-19,21]
   - report_two_digit_numbers([100,7,8437]) → []