

Gary Szekely

<https://github.com/garyszekelyjr/garyszekely-usdc-2024>

### **Overall Process & Decision Making:**

Before embarking on the problem-solving process, I prefer to gain a comprehensive understanding of the input parameters and desired output. Fundamentally, this task involves examining a list of books, determining whether a specified search term is present in their content, and subsequently identifying the locations where the term appears. Leveraging a high-level language such as JavaScript simplifies the implementation, with the primary challenge lying in ensuring the correctness of the output format.

Once I establish a basic working solution, my approach involves contemplating potential use cases for the function. This consideration aims to enhance the robustness of the code by accounting for diverse input scenarios. While a solution may flawlessly handle specific inputs, its efficacy can diminish when confronted with less-than-ideal input configurations. Consequently, I iterate on the code to address a variety of situations, thereby ensuring its adaptability and reliability.

In parallel, my unit testing strategy aligns with this philosophy, encompassing a broad spectrum of possible inputs. The objective is to rigorously validate the function, compelling it to produce valid results as a baseline requirement. This systematic approach not only guarantees the functionality of the code in various contexts but also contributes to its overall resilience and versatility.

### **Testing & Iteration:**

- i. My test cases are meticulously designed to bolster the robustness of the input handling within the function. My overarching objective for this project has been the development of a function that consistently produces valid results. To achieve this, I crafted tests that incorporate diverse test objects, varying in length, content presence, adherence to casing rules, and the potential for null inputs. This approach significantly contributes to code safety, mitigating the risk of system failures by ensuring the function consistently returns accurate results. In the pursuit of an even more comprehensive evaluation, given additional time, I would have preferred to include larger inputs in my test suite. Expanding beyond small examples of one or two books, this would have allowed for a more thorough assessment of the function's scalability and performance. This consideration underscores the importance of not only validating the correctness of the results but also assessing the efficiency and reliability of the function under different scales of input data.
- ii. I place considerable emphasis on the simplicity of my solution. Every component serves a purpose, ensuring that no parts remain unused. The code is designed with clarity in mind, facilitating a straightforward comprehension for anyone familiar with JavaScript.

- iii. The aspect of the problem that presented the greatest challenge was the unit testing phase. While the implementation of the actual function proved to be straightforward, particularly within the JavaScript language, crafting effective test cases introduced a level of complexity. Unlike the deterministic nature of the function itself, testing scenarios demanded a more open-ended approach, necessitating thoughtful consideration of potential use cases and corner cases.
- iv. I took measures to address potential edge cases where the inputs might not be valid. This proactive approach involved implementing checks and validations within the code to handle scenarios where input parameters deviate from the expected format or contain unexpected values.