Test Cases

1. The user is required to input the name of a disease in the provided text field. This input should be a string that corresponds to a recognized disease, such as "cold", "flu", or "headache".

2. The input data type expected is a string. It will represent the name of the disease entered by the user.

3. The function find Medicine() is essential for processing the user's input. It is triggered when the user clicks the "Find Medicine" button.

4. The input string is processed by first trimming any leading or trailing whitespace using trim() to ensure that unnecessary spaces do not affect the matching process.

5. The processed input is then converted to lowercase using LowerCase() to handle any case variations in the disease name, allowing for case-insensitive comparison.

6. The diseaseMedicineMap object, which is a JavaScript object, is used to store the mapping of disease names (as strings) to their corresponding medicines (also strings). This object is key to finding the medicine based on the disease input.

7. The function checks if the entered disease is a key in the diseaseMedicineMap object. If a match is found, the corresponding medicine is retrieved from the object.

8. If a matching disease is found, the program outputs the recommended medicine in a designated HTML div element.

9. If no match is found, the program displays a message saying, "Sorry, we don't have information on this disease."

10. The function relies on the assignment operator (=) to assign values to variables like disease and medicine, which store the processed input and the corresponding medicine.

11. The JavaScript string methods to LowerCase() and trim() are essential for normalizing user input, making it more robust and user-friendly by removing inconsistencies like extra spaces or different letter cases.

12. The program needs to handle different input formats, so it ensures that "Cold", "cold", and "COLD" are all treated the same by converting everything to lowercase.

13. The code currently does not handle diseases with special characters or multi-word diseases, meaning inputs like "headache" or "high blood pressure" will not return any result and will display the error message.

14. If the user leaves the input field empty, the program will also return the error message, as no disease name will match an empty string in the diseaseMedicineMap.

15. The overall purpose of the function is to provide a way for users to look up recommended medicines for specific diseases by entering the disease name, and to provide clear feedback for both valid and invalid inputs.

Algorithm

1.Start the Program:

The program is loaded in a web page where the user is prompted to enter the name of a disease.

2.Capture the user input:

- The program uses an HTML input field to capture the user's input.

- The input data type is string. The string represents the disease name entered by the user, such as "cold" or "flu".

3. Process the input:

- The trim() function is applied to the input string to remove any leading or trailing whitespace. This ensures that extra spaces do not interfere with the matching process.

- The to LowerCase() function is used on the string to convert all characters to lowercase, ensuring case-insensitivity (e.g., "Cold", "cold", and "COLD" are treated the same).

- After processing, the disease name is stored in the variable disease, which is a string.

4. Search for the disease:

- The processed disease name (now in lowercase) is checked against the keys of the diseaseMedicineMap object. This object is a JavaScript object that maps disease names (as strings) to their corresponding medicines (also strings).

- The program accesses the diseaseMedicineMap object using bracket notation, such as diseaseMedicineMap[disease], to find the corresponding medicine.

5. Check if the disease exists in the map:

- If the disease exists in the map (i.e., it is a valid key in the diseaseMedicineMap object), the corresponding medicine (a string) is retrieved.

- The medicine string is then stored in the variable medicine and displayed on the page. The message shown will be: "The recommended medicine for [disease] is [medicine]."

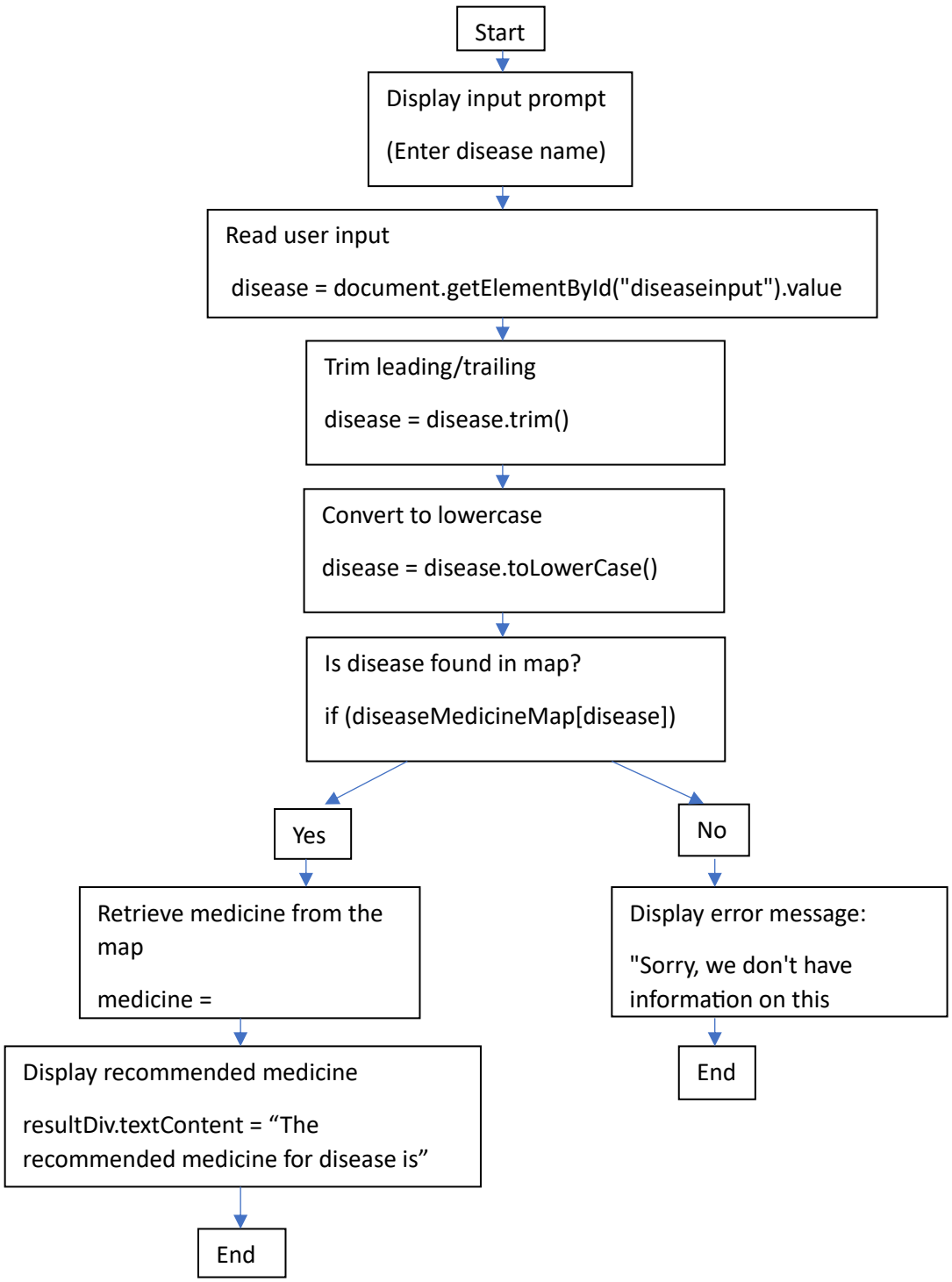
6. If the disease is not found:

- If the disease is not found in the map, the program displays an error message: "Sorry, we don't have information on this disease."

7. End the program:

- The program ends after displaying the result message, whether it's the recommended medicine or the error message.

Flowchart


Start
Display input prompt
(Enter disease name)
Read user input
disease = document.getElementById("diseaseinput").value
Trim leading/trailing
disease = disease.trim()
Convert to lowercase
disease = disease.toLowerCase()
Is disease found in map?
if (diseaseMedicineMap[disease])
Yes
No
Retrieve medicine from the map
medicine =
Display error message:
"Sorry, we don't have information on this
Display recommended medicine
resultDiv.textContent = “The recommended medicine for disease is”
End
End