To find a special character in a 2D std::vector<char> in C++, you can iterate through the rows and columns of the vector using nested loops. When you find the character, you can store its row and column indices and then stop the search.

# C++ Code Example

Here is a C++ function that takes a 2D vector of characters and a target character and returns the row and column indices. It returns (-1, -1) if the character isn't found.

C++

```
#include <iostream>
#include <vector>
#include <utility> // For std::pair

// Function to find a character in a 2D vector
std::pair<int, int> findChar(const std::vector<std::vector<char>>& grid, char target) {
    for (int i = 0; i < grid.size(); ++i) {
        for (int j = 0; j < grid[i].size(); ++j) {
            if (grid[i][j] == target) {
                return {i, j}; // Return row and column indices as a pair
            }
        }
    }
    return {-1, -1}; // Return a pair of -1s if not found
}

int main() {
    std::vector<std::vector<char>> myGrid = {
        {'a', 'b', 'c'},
        {'d', 'e', 'f'},
        {'g', 'h', 'i'}
    };

    char specialChar = 'e';
    std::pair<int, int> position = findChar(myGrid, specialChar);

    if (position.first != -1) {
        std::cout << "The special character '" << specialChar << "' was found at:" << std::endl;
        std::cout << "Row: " << position.first << std::endl;
```

```
        std::cout << "Column: " << position.second << std::endl;
    } else {
        std::cout << "The special character '" << specialChar << "' was not found." << std::endl;
    }

    return 0;
}
```

---

## Explanation of the Code

- **std::vector<std::vector<char>>**: This is a vector of vectors, which is a common way to represent a 2D array or grid in C++. The outer vector represents the rows, and each inner vector represents a column in that row.
- **Nested for loops**: The outer loop for (int i = 0; i < grid.size(); ++i) iterates through each **row**. The variable i represents the row index. The inner loop for (int j = 0; j < grid[i].size(); ++j) iterates through each **column** in the current row. The variable j represents the column index.
- **Comparison**: Inside the loops, if (grid[i][j] == target) checks if the character at the current row i and column j is the one we are looking for.
- **Return value**: The function uses std::pair<int, int> to return the row and column indices as a single object. This is a convenient way to return multiple values. If the character is found, it returns the indices. If the loops complete without finding the character, the function returns {-1, -1} to indicate that it wasn't found.
- **Output**: The main function demonstrates how to use findChar and prints the result. It checks the returned pair to see if the character was found before printing the location.