

Array and String Manipulation in C++ (Dynamic Input)

Objective:

Develop a C++ program to perform various operations on an array of names, where the user specifies how many names (`n`) to enter. Demonstrate the use of arrays and string manipulations.

Problem Statement

Write a program that:

1. Accepts `n` names from the user, where `n` is provided at runtime.
2. Performs the following operations:
 - Sorts the names alphabetically.
 - Searches for a specific name provided by the user (case-sensitive).
 - Identifies and displays the longest name in the list.
 - Counts and displays the total number of vowels (a, e, i, o, u) present in all names combined.
3. Displays the sorted list, search results, longest name, and vowel count.

Instructions

1. Input:

- Prompt the user to enter the number of names (`n`) they wish to input.
- Accept exactly `n` names from the user and store them in a dynamically allocated array of strings.

2. Processing:

- Perform the following operations:
 - Sorting: Sort the names alphabetically.
 - Searching: Allow the user to input a name to search for and indicate whether it exists in the list.
 - Longest Name: Find and display the longest name.
 - Vowel Count: Count and display the total number of vowels in all names.

3. Output:

- Display the sorted list of names.
- Display the result of the search operation.
- Display the longest name.
- Display the total count of vowels.

4. Constraints:

- Use a dynamically allocated array (`new string[n]`) to store the names.
- Implement the program using separate functions for each operation:
 - `void sortNames(string names[], int size)`
 - `bool searchName(const string names[], int size, const string& target)`
 - `string findLongestName(const string names[], int size)`
 - `int countVowels(const string names[], int size)`

5. Code Structure:

- Organize your code into:
 - Input: Collect user data.

- Processing: Perform operations.
- Output: Display the results.
- Use comments to explain the logic.

C++ Code: Dynamic Input with `n` Names

Grading Criteria

1. Correctness:

- All operations (sorting, searching, finding longest name, vowel counting) work correctly.

2. Dynamic Input:

- Handles any number of names (`n`) provided by the user.

3. Code Quality:

- Code is well-organized, with meaningful comments and proper formatting.

Let me know if you need further explanations! 😊

Sample output:

Enter the number of names: 5

Enter 5 names:

Enter name 1: Sumanth

Enter name 2: sumanth

Enter name 3: Burugula

Enter name 4: BURUGULA

Enter name 5: BuRUgUla

Sorted List of Names:

BURUGULA

BuRUgUla

Burugula

Sumanth

sumanth

Enter a name to search: BURUGULA

BURUGULA found in the list.

Longest Name: BURUGULA

Total Vowel Count: 16

Enter the number of names: 5

Enter 5 names:

Enter name 1: Chis

Enter name 2: Christian

Enter name 3: Matt

Enter name 4: Mathew

Enter name 5: Donald

Sorted List of Names:

Chis

Christian

Donald

Mathew

Matt

Enter a name to search: Donald

Donald not found in the list.

Longest Name: Christian

Total Vowel Count: 9

Enter the number of names: 1

Enter 1 names:

Enter name 1: Sandra

Sorted List of Names:

Sandra

Enter a name to search: SAndra

SAndra not found in the list.

Longest Name: Sandra

Total Vowel Count: 2