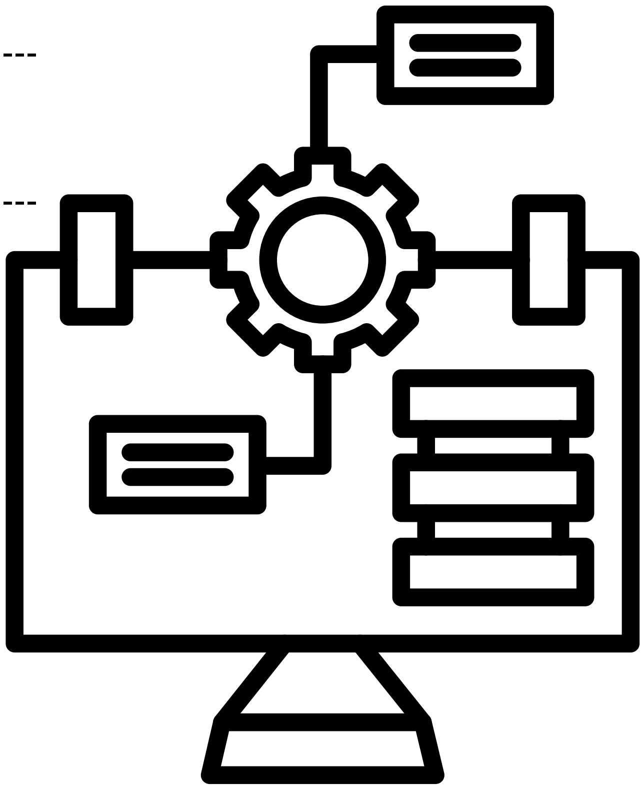
Report on All Functions in "Operations on Strings"

This report provides an overview of all the functions under the "Operations on Strings" section.

They are categorized into Basic, Intermediate, and Advanced operations.

----

1. Basic String Operations



1.1. Calculate String Length

Signature: int stringLength(char* str)

Description: Returns the length of the given string.

Example: For input "hello", the result is 5.

1.2. Copy String

Signature: void stringCopy(char* dest, const char* src)

Description: Copies the source string into the destination string.

1.3. Concatenate Strings

Signature: void stringConcat(char* dest, const char* src)

Description: Appends the source string to the end of the destination string.

Example: "hello" + "world" = "helloworld".

1.4. Compare Strings

Signature: int stringCompare(const char* str1, const char* str2)

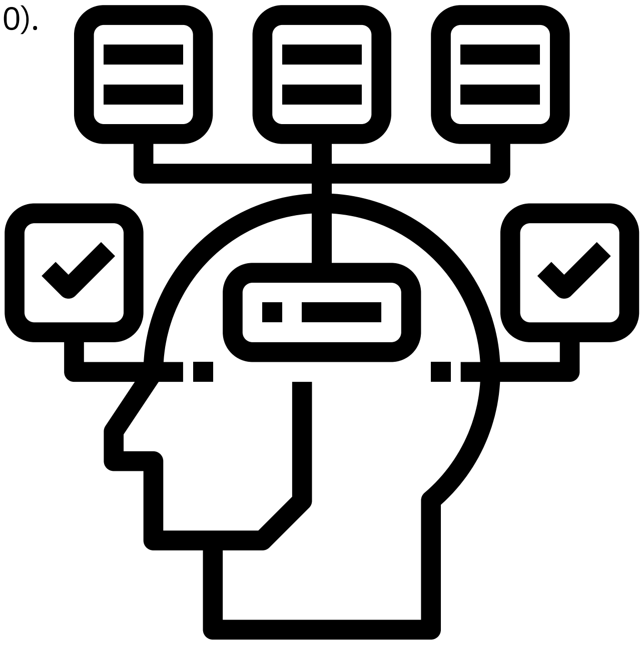
Description: Compares two strings lexicographically.

Example: "apple" < "banana".

1.5. Check if String is Empty

Signature: bool isEmpty(char* str)

Description: Checks if the given string is empty (length = 0).



1.6. Reverse a String

Signature: void reverseString(char* str)

Description: Reverses the characters in the string.

Example: "hello" becomes "olleh".

1.7. Convert to Uppercase

Signature: void toUpperCase(char* str)

Description: Converts all characters in the string to uppercase.

Example: "hello" becomes "HELLO".

1.8. Convert to Lowercase

Signature: void toLowerCase(char* str)

Description: Converts all characters in the string to lowercase.

Example: "HELLO" becomes "hello".

----

2. Intermediate String Operations

2.1. Palindrome Check

Signature: bool isPalindrome(char* str)

Description: Checks if the string reads the same forwards and backwards.

Example: "madam" is a palindrome.

2.2. Count Vowels and Consonants

Signature: void countVowelsConsonants(char* str, int* vowels, int* consonants)

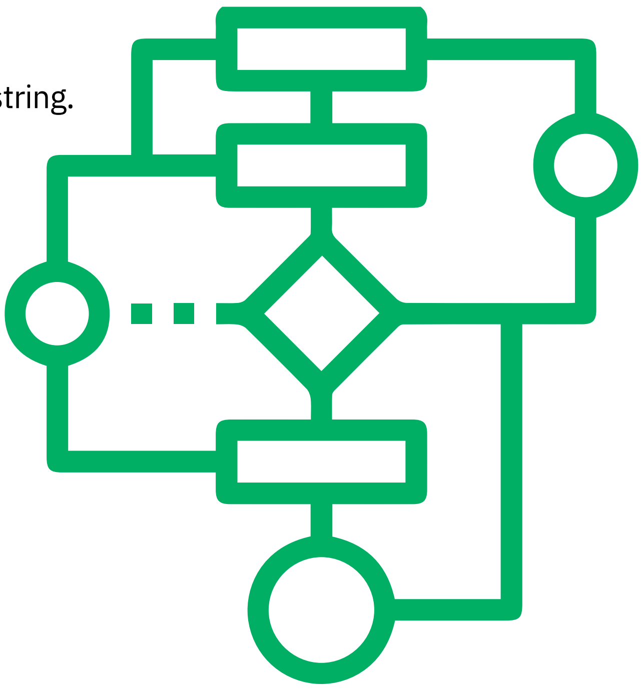
Description: Counts the number of vowels and consonants in the string.

2.3. Find Substring

Signature: int findSubstring(const char* str, const char* sub)

Description: Finds the first occurrence of a substring within a string.

Example: In "hello world", "world" starts at index 6.



2.4. Remove Whitespaces

Signature: void removeWhitespaces(char* str)

Description: Removes all spaces from the string.

Example: "hello world" becomes "helloworld".

2.5. Check Anagram

Signature: bool isAnagram(char* str1, char* str2)

Description: Checks if two strings are anagrams of each other.

Example: "listen" and "silent" are anagrams.

2.6. Character Frequency

Signature: void charFrequency(char* str, int* freq)

Description: Calculates the frequency of each character in the string.

2.7. Count Words

Signature: int countWords(char* str)

Description: Counts the number of words in the string.

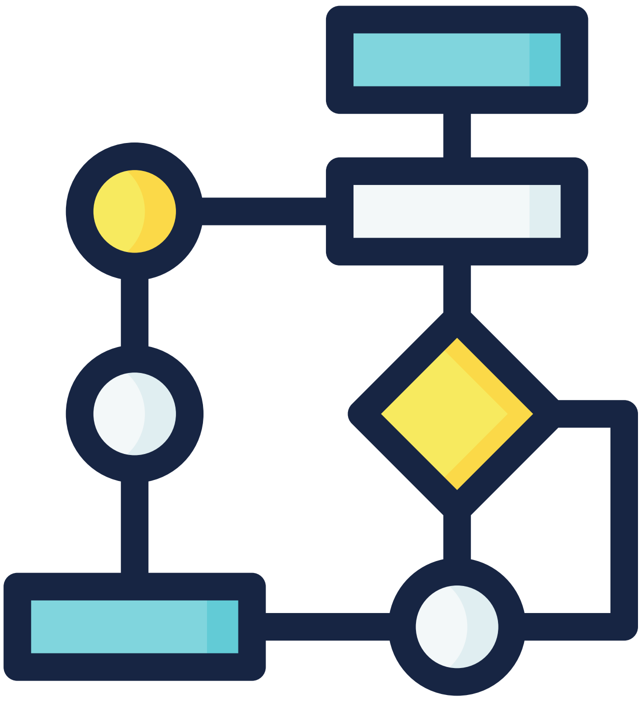
Example: "hello world" has 2 words.

2.8. Remove Duplicate Characters

Signature: void removeDuplicates(char* str)

Description: Removes duplicate characters from the string.

Example: "banana" becomes "ban".



----

3. Advanced String Operations

3.1. String Compression

Signature: void compressString(char* str, char* result)

Description: Compresses the string using Run-Length Encoding (RLE).

Example: "aaabbc" becomes "a3b2c1".

3.2. Find Longest Word

Signature: void longestWord(char* str, char* result)

Description: Finds the longest word in a sentence.

3.3. Check String Rotation

Signature: bool isRotation(char* str1, char* str2)

Description: Checks if one string is a rotation of another.

Example: "abc" is a rotation of "cab".

3.4. Count Specific Character

Signature: int countChar(char* str, char ch)

Description: Counts the occurrences of a specific character in the string.

Example: In "banana", 'a' appears 3 times.

3.5. Find and Replace Substring

Signature: void findAndReplace(char* str, char* find, char* replace)

Description: Replaces all occurrences of a substring with another substring.

Example: Replace "world" with "everyone" in "hello world" -> "hello everyone".

3.6. Find Longest Palindromic Substring

Signature: void longestPalindrome(char* str, char* result)

Description: Finds the longest palindromic substring in a given string.

Example: In "babad", the longest palindromic substring is "bab".

3.7. String Permutations

Signature: void printPermutations(char* str)

Description: Generates and prints all permutations of the string.

3.8. Split String

Signature: void splitString(char* str, char delimiter, char tokens[][100], int* tokenCount)

Description: Splits the string into tokens based on a specified delimiter.

Example: Splitting "hello,world" by ',' gives ["hello", "world"].

These functions form a comprehensive toolkit for string manipulation.

They are vital for tasks such as text processing, data analysis, and solving algorithmic problems.