

Implement SET operations on array line Assignment No. 2 without using built-in functions.

Roll No :

Date :

Problem Statement:

Write a menu driven C++ program with a class for String. Write functions

1. To determine the frequency of occurrence of a particular character in the string.
2. Extract a new string from original string by accepting starting position and length
3. To accept any character and return the string with by removing all occurrences of a character accepted
4. To make an in-place replacement of a substring w of a string by the string x. Note that w may not be of same size that of x
5. To check whether given string is palindrome or not

Objectives:

- Understand the concept of string to be implemented in C++.
- Understand the basic operation on given operations such as concatenation of String, find the length of string, string palindrome, deletion of substring & replacement of string using C++.

Outcomes:

Upon completion of assignment, student will be able to:

- 1) Design and identify how to use string concept in programming.
- 2) Understood the concept of string Structures

Theory

Here's an example of a string declaration in C++ :

```
char string[5];
```

Similar to arrays, you can also define strings as shown below :

```
char string[]="Welcome";
```

In this case, the array will hold the values "WLCOME and a null character '\0' that is added to the string at the end automatically by the compiler.

W	E	L	C	O	M	E	\0
---	---	---	---	---	---	---	----

Algorithm

To determine the frequency of occurrence of a particular character in the string.

1. Initialize a counter variable to store the count of total occurrences of a character in a string.
2. Traverse the string character by character.
3. If the character of the string matches with the given character, increment the value of the count variable.
4. Finally, return the counter variable

To accept any character and return the string with by removing all occurrences of a

Character accepted

1. Traverse the string character by character
2. If the character of the string does not matches with the given character print that character

Extract a new string from original string by accepting starting position and length

1. Read string
2. Read starting index of a string
3. Find length of the substring
4. Start traversing (print) the string from given index

To check whether string is palindrome or not

Step 1. Start

Step 2. Read the string from the user

Step 3. Calculate the length of the string

Step 4. Initialize rev = " " [empty string]

Step 5. Initialize i = length - 1

Step 6. Repeat until i>=0:

6.1: rev = rev + Character at position 'i' of the string

6.2: i = i - 1

Step 7. If string = rev:

7.1: Print "Given string is palindrome"

Step 8. Else:

8.1: Print "Given string is not palindrome"

Step 9. Stop

Conclusion

Thus we have studied the string operations using array

Practice problem:

1. Write a program to find the first and the last occurrence of the letter 'o' and character ',' in "Hello, World"
2. Write a program to enter a string s1 and copy it to another string s2.
3. Write the time complexity of all the functions of your code