॥वसुधैव कुटुम्बकम॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

Assignment No.: 09	
Course Name	Programming in C Lab
Name of Student	Faheemuddin Sayyed
PRN No.	23070122196
Branch	CSE
Class	C-1
Academic Year & Semester	2023-2024 & Semester 2
Date of Performance	05/04/2024
Assignment Title (Full):	Write a C program using functions to accept two strings from the console and perform the following operations (without using built-in functions): (a) Compare the strings: equal/not equal (b) Find the longer string. (c) Concatenate the two strings. (d) Find occurrence of substring in the main string

Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithm of the C program in the Journal Notebook, and add its screenshot in the below theory response.)

Theory Response:

- **Function Definitions:**
  1. length: Calculates the length of a string.
  2. Compare: Compares two strings for equality.
  3. Longer: Checks which string is longer.
  4. Concatenate: Concatenates two strings.
  5. Substring: Checks if one string contains another as a substring.
- **Main Function:**
  1. Input: Reads two strings (s1 and s2) and a substring (sub) from the user.
  2. Comparison: Checks if s1 and s2 are equal using Compare.
  3. Length Comparison: Determines which string (s1 or s2) is longer using Longer.
  4. Concatenation: Concatenates s2 onto s1 using Concatenate.
  5. Substring Check: Determines if s2 contains sub as a substring using Substring.

Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.)

Output Response:

```c
#include <stdio.h>

int length(char str[50]){
    int count=0;
    for(;str[count]!=0;count++);
    return count;
}

int Compare(char s1[50],char s2[50]){
    if(length(s1)!=length(s2)) return 0;
    for(int i=0;i<length(s1);i++){
        if(s1[i]!=s2[i]) return 0;
    }
    return 1;
}

int Longer(char s1[50], char s2[50]){
    if(length(s1)>length(s2)) return 1;
    return 0;
}

void Concatenate(char s1[50], char s2[50]){
    int l1=0,l2=0;
    l1=length(s1);
    l2=length(s2);
    for(int i=l1,j=0;i<l1+l2 && j<l2;i++,j++){
        s1[i]=s2[j];
    }
}

int Substring(char s1[50], char sub[50]){
    for(int i=0;i<length(s1);i++){
        for(int i=0;i<=length(s1)-length(sub);i++){
            int j;
            for(j=0;j<length(sub);j++){
                if(s1[i+j]!=sub[j]){
                    break;
                }
            }
            if(j==length(sub)) return 1;
        }
        return 0;
    }
    return 0;
}

int main(){
    char s1[50], s2[50],sub[50];
    printf("\nEnter 1st string: ");
    scanf("%[^\n]s",s1);
    getchar(); // Consume the newline character
    printf("\nEnter 2nd string: ");
    scanf("%[^\n]s",s2);
    getchar();
    Compare(s1,s2)?printf("\nStrings are equal.\n"):printf("\nStrings are not equal.\n");
    Longer(s1,s2)?printf("\n1st String is Longer.\n"):printf("\n2nd String is Longer.\n");
    Concatenate(s1,s2);
    printf("\nConcatenated string: %s\n",s1);
    printf("\nEnter substring to check in string 2:");
    scanf("%[^\n]s",sub);
    Substring(s2,sub)?printf("\nContains Substring.\n"):printf("\nDoes not contain substring.\n");
    return 0;
}
```

```
Enter 1st string: Hello World!

Enter 2nd string: Welcome to C

Strings are not equal.

2nd String is Longer.

Concatenated string: Hello World!Welcome to C

Enter substring to check in string 2:Welcome

Contains Substring.
(base) fahee@Faheems-MacBook-Pro Programming_in_C %
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

This program demonstrates basic string manipulation operations such as length calculation, comparison, concatenation, and substring checking using user-provided strings. It utilizes custom functions for these operations to encapsulate common tasks, promoting code reusability and clarity. However, it assumes string lengths won't exceed a certain limit (50 characters), which can be a limitation in practical scenarios.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte