	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

Title:	String Operations
Problem Statement	Write a Program in C to illustrate string operations without using library functions. Verify the operations using in-built functions. A. with pointers to arrays B. Without pointers to arrays

Programmer Name: Arpan Agrawal
Batch: E6

A. String operations without pointers

```
#include<stdio.h>

// Function to find the length of a string
int stringLength(char str[]) {
    int length = 0;
    while (str[length] != '\0') {
        length++;
    }
    return length;
}

// Function to find substring
int substring(char str[], char substr[]) {
    int i, j;
    int len_str = stringLength(str);
    int len_substr = stringLength(substr);

    for (i = 0; i <= len_str - len_substr; i++) {
        j = 0;
```

```
        while (str[i + j] == substr[j]) {
            j++;
            if (j == len_substr) {
                return i;
            }
        }
    }
    return -1; // Substring not found
}

// Function to check if string is a palindrome
int isPalindrome(char str[]) {
    int i, j;
    int len = stringLength(str);
    for (i = 0, j = len - 1; i < j; i++, j--) {
        if (str[i] != str[j]) {
            return 0; // Not palindrome
        }
    }
    return 1; // Palindrome
}

// Function to compare two strings
int compareStrings(char str1[], char str2[]) {
    int i = 0;
    while (str1[i] != '\0' && str2[i] != '\0') {
        if (str1[i] < str2[i]) {
            return -1; // str1 is smaller
        }
        if (str1[i] > str2[i]) {
            return 1; // str2 is smaller
        }
        i++;
    }
    // If both strings end at the same time
```

```
    if (str1[i] == '\0' && str2[i] == '\0') {
        return 0; // Both strings are equal
    }
    if (str1[i] == '\0') {
        return -1; // str1 is smaller
    }
    return 1; // str2 is smaller
}

// Function to copy one string to another
void copyString(char source[], char destination[]) {
    int i = 0;
    while (source[i] != '\0') {
        destination[i] = source[i];
        i++;
    }
    destination[i] = '\0'; // Add null character at the end
}

// Function to reverse a string
void reverseString(char str[]) {
    int i, j;
    char temp;
    int len = stringLength(str);
    for (i = 0, j = len - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

int main() {
    char str[100], substr[100];
    int choice;
    printf("Enter a string: ");
```

```
scanf("%s", str);
printf("Choose an operation-\n");
printf("1. Substring\n");
printf("2. Palindrome\n");
printf("3. Compare\n");
printf("4. Copy\n");
printf("5. Reverse\n");

printf("Enter your choice: ");
scanf("%d", &choice);
char copy[100];
char str2[100];

switch (choice) {
   case 1:
      printf("Enter substring to search: ");
      scanf("%s", substr);
      int index = substring(str, substr);
      if (index != -1) {
         printf("Substring found at index: %d\n", index);
      } else {
         printf("Substring not found\n");
      }
      break;

   case 2:
      if (isPalindrome(str)) {
         printf("The string is a palindrome.\n");
      } else {
         printf("The string is not a palindrome.\n");
      }
      break;

   case 3:
      printf("Enter another string: ");
```

```
            scanf("%s", str2);
            int result = compareStrings(str, str2);
            if (result < 0) {
                printf("The first string is smaller.\n");
            } else if (result > 0) {
                printf("The second string is smaller.\n");
            } else {
                printf("Both strings are equal.\n");
            }
            break;

        case 4:
            copyString(str, copy);
            printf("Copied string: %s\n", copy);
            break;

        case 5:
            reverseString(str);
            printf("Reversed string: %s\n", str);
            break;

        default:
            printf("Invalid choice. Please choose a valid operation.\n");
    }
    return 0;
}
```