Batch: C1-2	**Roll No.: 16010123036**
Experiment / assignment / tutorial No. 5	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program in C to demonstrate use of character arrays and strings

AIM:

a) Write a program that searches for a substring within a given string.
b) Write a program to check if one string is the rotation of another.

Expected OUTCOME of Experiment:

Apply the concepts of arrays and strings(CO3).

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.

Problem Definition:

1. The program searches for a substring within a given string and returns the starting index if found, or -1 otherwise.

Example:

Test case 1: Input: String: Programming Substring: ing Output:	Test case 2: Input: String: Programming Substring: Python Output:

Index: 8	Index: -1

2. The program checks whether a given string is the rotation of the other.

Example:

Test case 1: Input: String 1: abcd String 2: bcda Output: Yes	Test case 2: Input: String 1: abcd Substring: dcba Output: No

Algorithm:

1)

1. Start

2. Declare character arrays 'str' and 'substr' of size 100 to store the string and substring respectively.

3. Prompt the user to enter the string and read it into 'str'.

4. Prompt the user to enter the substring and read it into 'substr'.

5. Calculate the length of 'str' and 'substr' using strlen() function and store them in 'len_str' and 'len_substr' respectively.

6. Declare an integer variable 'index' and initialize it to -1.

7. Implement a nested loop to search for the substring within the string:

a. Iterate over the characters of the string using the outer loop (i) from 0 to (len_str - len_substr).

b. Initialize an inner loop (j) to iterate over the characters of the substring.

c. Inside the inner loop, check if the characters of 'str' starting from index i match the characters of 'substr'.

d. If any character doesn't match, break out of the inner loop.

e. If all characters of 'substr' match, set 'index' to the starting index 'i' and break out of the outer loop.

8. If 'index' is not equal to -1, print the index where the substring is found.

9. If 'index' is -1, print -1 indicating that the substring is not found.

10. End

2)

1. Start

2. Declare two character arrays 'str1' and 'str2' of size 100 to store two strings.

3. Prompt the user to enter String 1 and read it into 'str1'.

4. Prompt the user to enter String 2 and read it into 'str2'.

5. Calculate the lengths of 'str1' and 'str2' using strlen() function and store them in 'len1' and 'len2' respectively.

6. If the lengths of 'str1' and 'str2' are not equal or 'str1' is empty (length is 0):

   a. Print "No".

   b. Exit the program.

7. Declare a character array 'temp' of size 2 times 'len1' + 1 to concatenate 'str1' with itself.

8. Copy 'str1' into 'temp' using strcpy() function.

9. Concatenate 'str1' with itself in 'temp' using strcat() function.

10. Check if 'str2' is a substring of 'temp' using strstr() function:

   a. If 'str2' is found in 'temp':

      i. Print "Yes".

   b. Else:

      i. Print "No".

11. End

Implementation details:

1)

```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[100], substr[100];
    printf("Enter the string: ");
    scanf("%s", str);
    printf("Enter the substring: ");
    scanf("%s", substr);

    int len_str = strlen(str);
    int len_substr = strlen(substr);
    int index = -1;

    for (int i = 0; i <= len_str - len_substr; ++i) {
        int j;
        for (j = 0; j < len_substr; ++j) {
            if (str[i + j] != substr[j]) {
                break;
            }
        }
        if (j == len_substr) {
            index = i;
            break;
        }
    }

    if (index != -1) {
        printf("Index: %d\n", index);
    } else {
        printf("Index: -1\n");
    }

    return 0;
}
```

2)

```c
#include <stdio.h>
#include <string.h>

int main() {
    char str1[100], str2[100];
    printf("Enter String 1: ");
    scanf("%s", str1);
    printf("Enter String 2: ");
    scanf("%s", str2);

    int len1 = strlen(str1);
    int len2 = strlen(str2);

    if (len1 != len2 || len1 == 0) {
        printf("No\n");
        return 0;
    }

    char temp[2 * len1 + 1];
    strcpy(temp, str1);
    strcat(temp, str1);

    if (strstr(temp, str2) != NULL)
        printf("Yes\n");
    else
        printf("No\n");

    return 0;
}
```

Output(s):

1)

```
Enter the string: Programming
Enter the substring: ing
Index: 8
```

```
Enter the string: Programming
Enter the substring: Python
Index: -1
```

2)

```
Enter String 1: abcd
Enter String 2: bcda
Yes
```

```
Enter String 1: abcd
Enter String 2: dcba
No
```

Conclusion:

The experiments showcased the practical use of character arrays and strings in C. We created functions to find substrings within strings and determine if one string is a rotation of another. These exercises deepened our understanding of string manipulation using array operations and C library functions. They underscored the vital role of arrays and strings in programming, demonstrating their effectiveness in solving diverse problems.

Post Lab Questions

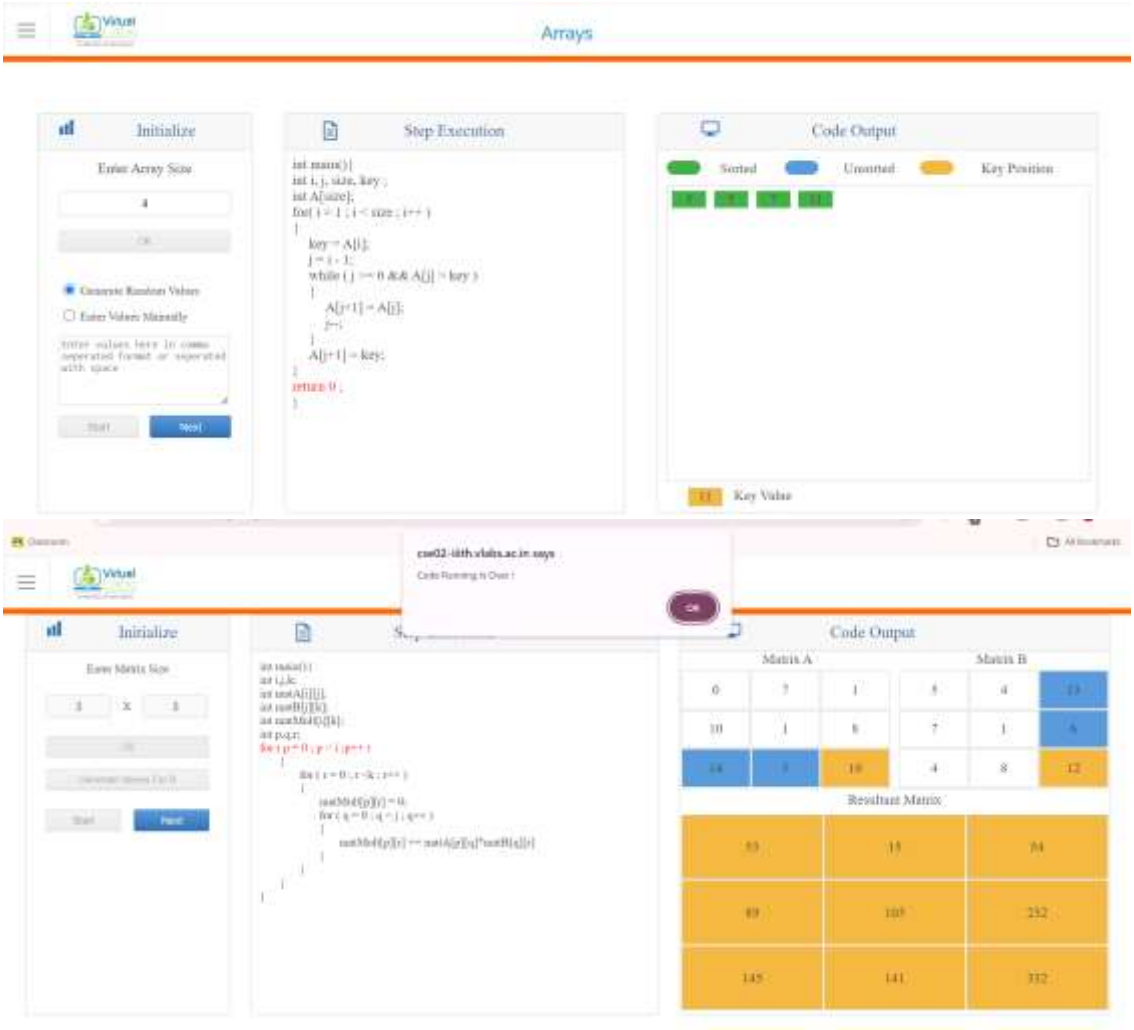
1. Write a C program to toggle case of each character in a string i.e. if a character is in uppercase, change it to lower case and vice-versa.

Ans:

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char str[100];

    printf("Enter a string: ");
    scanf("%[^\n]s", str);

    for (int i = 0; str[i] != '\0'; ++i) {
        if (islower(str[i])) {
            str[i] = toupper(str[i]);
        } else if (isupper(str[i])) {
            str[i] = tolower(str[i]);
        }
    }
    printf("Modified string: %s\n", str);

    return 0;
}
```

```
Enter a string: ProGramming
Modified string: pROgRAMMING
```

2. Virtual Lab for Arrays
   https://cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html



Date: _____ **Signature of faculty in-charge**