

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

**AIM:**

1. Write a program that searches for a substring within a given string.
2. 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:  Index: 8 | Test case 2:  Input:  String: Programming  Substring: Python  Output:  Index: -1 |
| --- | --- |

1. 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. **We begin by including the necessary libraries, which provide functions for input/output and string manipulation.**
2. **In the main function, we first print our name and roll number using printf.**
3. **We then declare two arrays of characters str and substr to store the main string and the substring.**
4. **Next, we declare variables i, j, k, len1, len2, and found to be used in the program.**
5. **We prompt the user to input the main string and the substring.**
6. **We calculate the lengths of the main string and the substring using the strlen function.**
7. **We use a loop to iterate through the characters of the main string to check if the substring is a rotation of it.**
8. **Inside the loop, we compare the characters of the main string and the substring to check for a match.**
9. **If a match is found, we print the index at which the substring starts in the main string. If no match is found, we print -1.**

**2)**

1. **The code includes header files such as stdio.h, stdlib.h, and string.h which are used for input and output operations, memory allocation, and string manipulations, respectively.**
2. **In the main function, three character arrays s1, s2, and s3 are declared to store the input strings and their concatenated version.**
3. **The program then prompts the user to enter the main string and the sustring using printf and scanf functions.**
4. **It then uses strcpy and strcat functions to create a concatenated string s3 by combining the main string s1 with itself.**
5. **The code then uses nested loops to compare the characters of the concatenated string s3 with the substring s2 to check if it is a rotation. If a rotation is found, it sets the index variable to the position of the rotation.**
6. **Finally, it prints "Yes" if a rotation is found, and "No" if not.**

**Implementation details:**

1)

#include <stdio.h>

int main() {

char mainString[100], subString[50];

int mainLength, subLength, i, j, found;

printf("Enter the main string: ");

gets(mainString);

printf("Enter the substring to search: ");

gets(subString);

mainLength = strlen(mainString);

subLength = strlen(subString);

found = -1;

for (i = 0; i <= mainLength - subLength; i++) {

for (j = 0; j < subLength; j++) {

if (mainString[i + j] != subString[j]) {

break;

}

}

if (j == subLength) {

found = i;

break;

}

}

printf("Substring found at index: %d\n", found);

return 0;

}

2)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char s1[30];

char s2[30];

char s3[60];

int i,j,index,t;

printf("Akshat Bachhuka 16010123034 \n");

printf("Enter MainString: ");

scanf("%s",&s1);

printf("Enter Substring: ");

scanf("%s",&s2);

strcpy(s3,s1);

strcat(s3,s1);

for(i=0;i<=strlen(s1);i++)

{

t=i;

for(j=0;j<=strlen(s2);j++)

{

if(s3[t]==s2[j])

{

index=i;

}

else

{

index=-1;

}

t++;

}

if(index!=-1)

{

break;

}

}

printf("\n");

if(index!=-1)

{

printf("Yes");

}

else

{

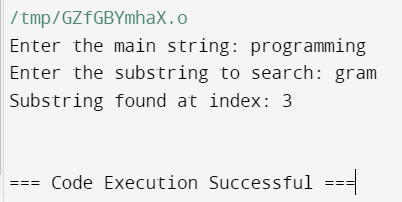
printf("No");

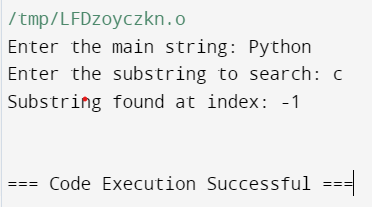
}

return 0**;**

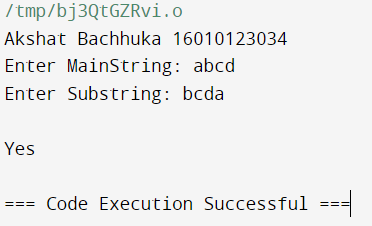
**}**

**Output(s):**





2)



**Conclusion:**

We learnt about versatile uses of arrays and strings and their functions respectively.

**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.

#include <stdio.h>

int main() {

char inputString[100];

printf("Akshat Bachhuka\n 16010123034: \n");

printf("Enter a string: ");

gets(inputString);

for (int i = 0; inputString[i] != '\0'; i++) {

if (inputString[i] >= 'A' && inputString[i] <= 'Z') {

inputString[i] = inputString[i] + 32; // Convert uppercase to lowercase

} else if (inputString[i] >= 'a' && inputString[i] <= 'z') {

inputString[i] = inputString[i] - 32; // Convert lowercase to uppercase

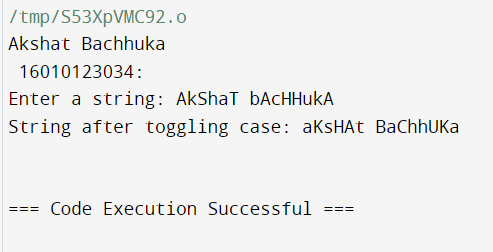
}

}

printf("String after toggling case: %s\n", inputString);

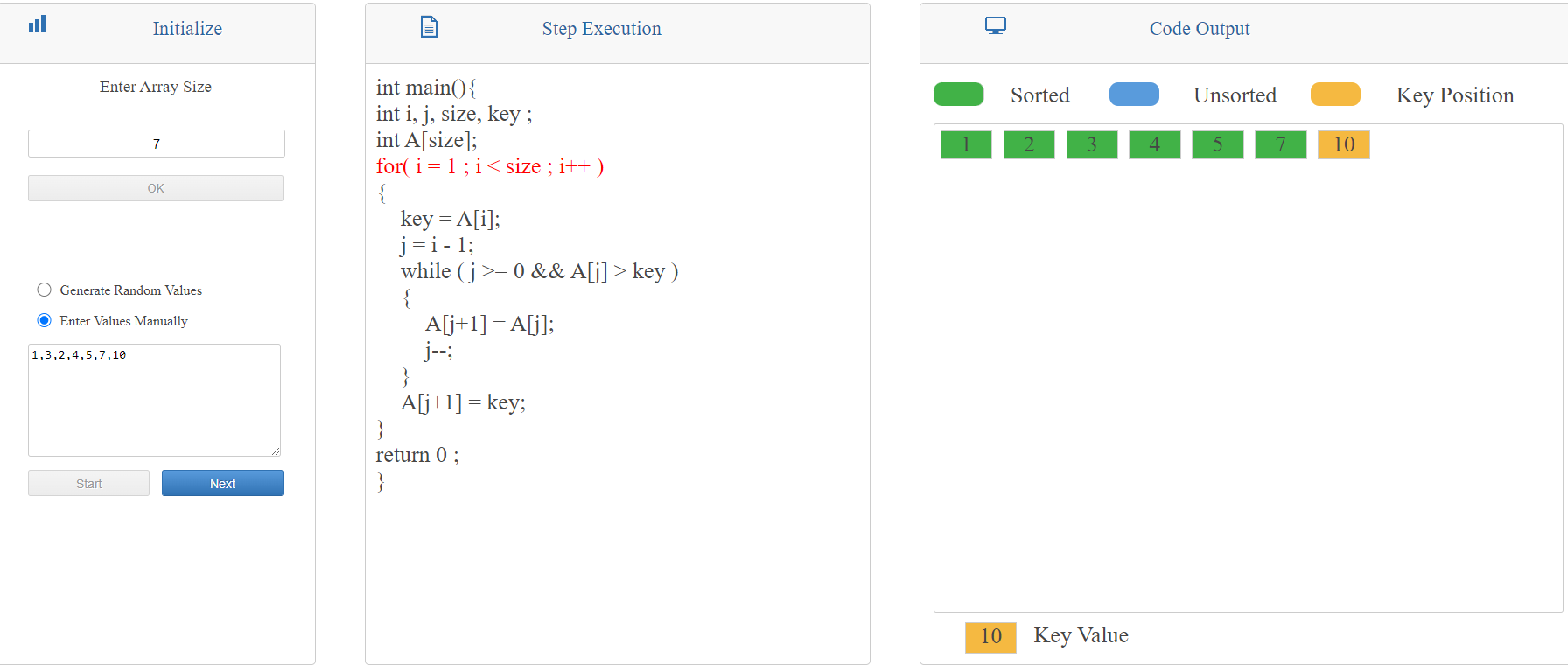
return 0;

}



1. Virtual Lab for Arrays

<https://cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html>



**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**