Batch: P4-1 **Roll No.: 16010423076**

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: Index: 8	Test case 2: Input: String: Programming Substring: Python Output: 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)

Calculate the lengths of the two input strings (len1 for s1 and len2 for s2).

Iterate through s2 from index 0 to len2 - len1.

For each starting index i, iterate through both strings character by character.

Inside the inner loop, compare the characters of s1 and the corresponding characters of the substring of s2 starting at index i.

If at any point the characters don't match, break out of the inner loop.

If the inner loop completes (i.e., all characters match), return the current index i as the starting index of the substring.

If the outer loop completes without finding a match, return -1 to indicate that s1 is not a substring of s2.

2)

Declare two constant character arrays s1 and s2 and initialize them with "abcd" and "bcda" respectively.

Declare two integer variables len1 and len2 and assign the lengths of s1 and s2 using the strlen function.

Check if len1 is equal to len2 and greater than 0.

i. If true, proceed to the next step.

ii. If false, jump to the end of the program.

Declare a character array temp with a size of 2 * len1 + 1.

Copy the content of s1 to temp using the strcpy function.

Concatenate the content of s1 to temp using the strcat function.

Check if s2 is a substring of temp using the strstr function.

i. If true, print "Yes" to the console.

ii. If false, print "No" to the console.

End the main function.

Implementation details:

1)

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


int sscheck(char* s1, char* s2)
{
int strl1 = strlen(s1);
int strl2 = strlen(s2);

for (int i=0;i<=strl2-strl1;i++) {
        int j;

        for(j=0;j<strl1;j++)
                if (s2[i+j] != s1[j])
                        break;
        if (j == strl1)
                return i;
}
return -1;
}
```

Test Case 1:

```
Void main()
{
char s1[] = "ing";
char s2[] = "Programming";
int indloc = sscheck(s1,s2);
if (indloc == -1)
        printf("It is not a substring");
else
        printf("It is a substring present at index %d",indloc);
}
```

Test Case 2:

```
void main()
{
char s1[] = "Python";
char s2[] = "Programming";
```

```
int indloc = sscheck(s1,s2);
if (indloc == -1)
        printf("It is not a substring");
else
        printf("It is a substring present at index %d",indloc);
}
```

2)

Test Case 1:

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

int main() {
   const char *s1 = "abcd";
   const char *s2 = "bcda";

   int len1 = strlen(s1);
   int len2 = strlen(s2);

   if(len1 == len2 && len1 > 0){
      char temp[2*len1+1];
      strcpy(temp,s1);
      strcat(temp,s1);

      if(strstr(temp,s2) != NULL) {
         printf("Yes");
      } else {
         printf("No");
      }
   }
   return 0;
}
```

Test Case 2:

```
const char *s1 = "abcd";
const char *s2 = "dcba";
```

Output(s):

1)Test Case 1:

```
It is a substring present at index 8

...Program finished with exit code 0
Press ENTER to exit console.
```

Test Case 2:

```
It is not a substring

...Program finished with exit code 0
Press ENTER to exit console.
```

2)Test Case 1:

```
Yes

...Program finished with exit code 0
Press ENTER to exit console.
```

Test Case 2:

```
No

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion:

Through these programs, I gained a basic understanding of working with strings in C. The substring search program involved iterating through characters to find a match, returning the starting index if successful. The string rotation check program used concatenation and substring comparison to determine if one string is a rotation of another. These exercises provided insight into fundamental string manipulation concepts in C programming.

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.

Input:

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

void convert(char* str)
{
int length = strlen(str);
for(int i=0;i<length;i++) {
        if (str[i]>='a' && str[i]<='z')
                str[i] = str[i]-32;
        else if (str[i] >= 'A' && str[i] <= 'Z')
                str[i] = str[i]+32;
}
}

void main()
{
char str[] = "My Name is RiTeSH.";
convert(str);
printf("%s", str);
}
```

Output:

```
mY nAME IS rItEsh.

...Program finished with exit code 0
Press ENTER to exit console.
```

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

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