

Name	Lekh Sanatan Nayak
UID no.	2023800068
Experiment No.	7

AIM:	Implement various text processing problems
Program 1	
PROBLEM STATEMENT :	Write a program to count the number of vowels, consonants, total characters and words in the given string.
ALGORITHM:	<ol style="list-style-type: none"> 1. Start 2. Include libraries such as “stdio.h” and “ctype.h” 3. Declare a function “main” 4. Define 6 integers variables ‘consonents’, ‘digits’, ‘vowels’, ‘spaces’, ‘character’, ‘words’ and initialize each of them to 0 5. Define an array “str” and store “Lekh Nayak 69” in it 6. Call the library function strlen() with arguments ‘str’ and store the value of length of the string ‘str’ in integer variable ‘len’ 7. Create a for loop with the loop variable ‘i’ and iterate form 0 to ‘len-1’ 8. Check if the character is a digit using “isdigit()” library function and increment the ‘digits’ variable if true 9. Check if the character is a consonant i.e. its not a vowel or space or digits and increment the ‘consonents’ variable if true 10. Check if the character is a vowel i.e. a, e ,i ,o ,u and increment the ‘vowels’ variable if true 11. Check if the character is a space and increment the ‘space’ variable if true 12. Increment the ‘character’ variable for each character 13. Check if the previous character was a space to identify the start of the word and increment the word count accordingly 14. Print the counts of consonents, vowels, digits, spaces, characters and words 15. End

PROGRAM:

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

int main()
{
    int consonents = 0, digits = 0, vowels = 0, spaces = 0, character = 0,
words = 0;
    char str[] = "Lekh Nayak 69";
    int len = strlen(str);

    for (int i = 0; i < len; i++)
    {
        str[i] = tolower(str[i]);
        if (isdigit(str[i]))
        {
            digits++;
        }
        else if (str[i] != 'a' && str[i] != 'e' && str[i] != 'i' && str[i] !=
'o' && str[i] != 'u' && str[i] != ' ')
        {
            consonents++;
        }
        else if (str[i] != ' ')
        {
            vowels++;
        }
        else
        {
            spaces++;
        }
        character++;
        if (i == 0 && str[i] != ' ' || str[i] != ' ' && str[i - 1] == ' ')
        {
            words++;
        }
    }

    printf("Consonants: %d\nVowels: %d\nDigits: %d\nSpaces:
%d\nCharacters: %d\nWords: %d", consonents, vowels, digits, spaces,
character, words);
```

	}
--	---

RESULT:

```
cyclops@cyclops:~/Downloads/woo$ gcc strings.c
cyclops@cyclops:~/Downloads/woo$ ./a.out
Consonants: 6
Vowels: 3
Digits: 2
Spaces: 2
Characters: 13
Words: 3cyclops@cyclops:~/Downloads/woo$
```

Program 2

PROBLEM STATEMENT :	<p>Write a Menu driven Program to</p> <ul style="list-style-type: none"> i) copy one string to another one by one character. ii) Find the string length iii) compare two strings iv) reverse the string v) Concatenate one string to another string. vi) lower case to upper
ALGORITHM:	<p>Algorithm for Copy string function (with 2 'char' arguments 'src' and 'dest')</p> <ol style="list-style-type: none"> 1. Initialize index 'I' to 0 2. Create a while loop while character at index 'i' in the source string is not the null terminator '\0' 3. Copy the character at index 'I' from the source string to the

- destination string
4. Increment 'i'
 5. Add the null terminator '\0' to the end of the destination string

Algorithm for stringLength function (with 1 char argument 'str')

1. Initialize length to 0.
2. While the character at index length in the string is not the null terminator '\0':
3. Increment length.
4. Return the calculated length.

Algorithm for compareStrings function (with 2 'char' arguments 'str1' and 'str2')

1. Initialize index i to 0.
2. While either the character at index i in the first string or the character at index i in the second string is not the null terminator '\0':
3. If the characters at index i in both strings are not equal, return 0.
4. Increment i.
5. If both strings reach the null terminator at the same time, return 1. Otherwise, return 0.

Algorithm for reverseString function (with 1 char argument 'str')

1. Find the length of the string.
2. Initialize indices i to 0 and j to length - 1.
3. While i is less than j:
4. Swap the characters at indices i and j.
5. Increment i and decrement j.

Algorithm for concatenateStrings function (with 2 'char' arguments 'src' and 'dest')

1. Find the length of the destination string.
2. Initialize index i to the length of the destination string and index j to 0.
3. While the character at index j in the source string is not the null terminator '\0':
4. Copy the character at index j from the source string to the

	<p>destination string at index i.</p> <ol style="list-style-type: none"> Increment both i and j. Add the null terminator '\0' to the end of the destination string. <p>Algorithm for lowercasetoUpper function (with 1 char argument 'str')</p> <ol style="list-style-type: none"> Initialize index i to 0. While the character at index i in the string is not the null terminator '\0': If the character at index i is a lowercase letter, convert it to uppercase. Increment i. <p>Algorithm for Menu-driven Program (i.e. main function)</p> <ol style="list-style-type: none"> Display the menu options. Prompt the user to enter a choice. Based on the choice: Call the corresponding function with user input if required. Print the result or an error message if the choice is invalid.
PROGRAM:	<pre>#include<stdio.h> void copyString(char src[], char dest[]){ int i = 0; while (src[i] != '\0') { dest[i] = src[i]; i++; } dest[i] = '\0'; } int stringLength(char str[]) { int length = 0; while (str[length] != '\0') { length++; } return length; }</pre>

```

int compareStrings(char str1[], char str2[]) {
    int i = 0;
    while (str1[i] != '\0' || str2[i] != '\0') {
        if (str1[i] != str2[i]) {
            return 0;
        }
        i++;
    }
    if (str1[i] == '\0' && str2[i] == '\0') {
        return 1;
    }
    else {
        return 0;
    }
}

void reverseString(char str[]) {
    int length = stringLength(str);
    int i, j;
    for (i = 0, j = length - 1; i < j; i++, j--) {
        char temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

void concatenateStrings(char dest[], char src[]) {
    int i = stringLength(dest);
    int j = 0;
    while (src[j] != '\0') {
        dest[i] = src[j];
        i++;
        j++;
    }
    dest[i] = '\0';
}

void lowercaseToUpper(char str[]) {
    int i = 0;
    while (str[i] != '\0') {

```

```

        if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] = str[i] - 32;
        }
        i++;
    }
}

int main(){
    int menu;
    char str1[100], str2[100];

    printf("Menu:-\n");
    printf("1. Copy one string to another string:-\n");
    printf("2. Find the string length\n");
    printf("3. Compare two strings\n");
    printf("4. Reverse the string\n");
    printf("5. Concatenate one string to another string\n");
    printf("6. Convert lower case to upper case\n");
    printf("Enter your choice:-");
    scanf("%d",&menu);

    switch (menu)
    {
    case 1 :
        printf("Enter the source string: ");
        scanf("%s", str1);
        copyString(str1, str2);
        printf("Copied string: %s\n", str2);
        break;

    case 2 :
        printf("Enter the string: ");
        scanf("%s", str1);
        printf("Length of the string: %d\n", stringLength(str1));
        break;

    case 3 :
        printf("Enter the first string: ");
        scanf("%s", str1);

```

```

printf("Enter the second string: ");
scanf("%s", str2);
if (compareStrings(str1, str2) == 1) {
    printf("The strings are equal.\n");
}
else {
    printf("The strings are not equal.\n");
}
break;

case 4 :
    printf("Enter the string: ");
    scanf("%s", str1);
    reverseString(str1);
    printf("Reversed string: %s\n", str1);
    break;

case 5 :
    printf("Enter the first string: ");
    scanf("%s", str1);
    printf("Enter the second string: ");
    scanf("%s", str2);
    concatenateStrings(str1, str2);
    printf("Concatenated string: %s\n", str1);
    break;

case 6 :
    printf("Enter the string in lowercase: ");
    scanf("%s", str1);
    lowercaseToUpper(str1);
    printf("Uppercase string: %s\n", str1);
    break;

default:
    printf("Invalid choice. Please try again.\n");
    break;
}

return 0;
}

```

RESULT:


```
cyclops@cyclops:~/Downloads/woo$ gcc string.c
cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-1
Enter the source string: lekh
Copied string: lekh
cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-2
Enter the string: lekhnayak
Length of the string: 9
cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-3
Enter the first string: lekh
Enter the second string: nayak
The strings are not equal.
cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-4
Enter the string: lekh
Reversed string: hkel
```

```

cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-5
Enter the first string: lekh
Enter the second string: nayak
Concatenated string: lekhnayak
cyclops@cyclops:~/Downloads/woo$ ./a.out
Menu:-
1. Copy one string to another string:-
2. Find the string length
3. Compare two strings
4. Reverse the string
5. Concatenate one string to another string
6. Convert lower case to upper case
Enter your choice:-6
Enter the string in lowercase: lekhnayak
Uppercase string: LEKHNAYAK

```

Program 3

PROBLEM STATEMENT:

Write a program which reads a piece of text and outputs any palindromes that it contains.

Input: I AM SURE THE DEED IS ON THE LEVEL MADAM

Output: I DEED LEVEL MADAM

ALGORITHM:

1. Start
2. Include libraries such as "stdio.h" "string.h" "ctype.h"
3. Declare a function "isPalindrome" with arguments char str[], int start, int end
4. Initialize a while loop that continues when 'start < end'
5. In the loop compare the characters at the start and end after converting them to lowercase by using library function "tolower"
6. If the characters are not same return 0, showcasing the word is not a palindrome
7. If the characters are same, increment the start index and decrement the end index by one.
8. If the loop completes itself showcasing there are no non matching characters, return 1, showing the word is a palindrome
9. Declare a function "main"

	<ol style="list-style-type: none"> 10. Define an array 'inputtext' to store the sentence upto 100 characters 11. Prompt the user to enter a sentence and read it using fgets 12. Determine the length of the string using library function "strlen" 13. Define two variables 'wordStart' and 'wordEnd' to kkeep track of the start and end indices of words 14. Loop through each character in the input sentence and check if the character is a space or a newline character, which indicates the end of a word 15. If it is the end of the word, set 'wordEnd' to the previous character's index i.e. (i-1) 16. Check if the word from 'wordStart' to 'wordEnd' is a palindrome using the "isPalindrome" function 17. If it is a palindrome, orint the word character by character 18. Continue this process untill all the words are check if they are palindrome or not 19. Print the palindromic words 20. End
PROGRAM:	<pre> #include <stdio.h> #include <string.h> #include <ctype.h> int isPalindrome(char str[], int start, int end) { while (start < end) { if (tolower(str[start]) != tolower(str[end])) { return 0; } start++; end--; } return 1; } int main() { char inputtext[100]; printf("Enter a sentence- "); </pre>

```

fgets(inputText, 100, stdin);
printf("Words that are palindromes- ");
int textLength = strlen(inputText);
int wordStart = 0;
int wordEnd = 0;
for (int i = 0; i < textLength; i++)
{
    if (inputText[i] == ' ' || inputText[i] == '\n')
    {
        wordEnd = i - 1;
        if (isPalindrome(inputText, wordStart, wordEnd))
        {
            for (int j = wordStart; j <= wordEnd; j++)
            {
                printf("%c", inputText[j]);
            }
            printf("\t");
        }
        wordStart = i + 1;
    }
}

return 0;
}

```

RESULT:

```

cyclops@cyclops:~/Downloads/woo$ gcc strings.c
cyclops@cyclops:~/Downloads/woo$ ./a.out
Enter a sentence- the deed is on the level madam
Words that are palindromes- deed        level    madam

```

CONCLUSION:

In this experiment I learnt how to solve various text processing problems using Strings