Name	Jhaveri Varun Nimitt
UID no.	2023800042
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:	1. Start 2. Initialize variables to store the counts for consonants, vowels, digits, spaces, characters, and words. 3. Initialize the string with the given text. 4. Calculate the length of the string. 5. Convert the string to lowercase for consistent vowel and consonant counting. 6. Iterate over each character in the string: a. Check if the current character is a digit. i. If it is, increment the digits counter. b. Check if the current character is not a vowel (a, e, i, o, u) or a space. i. If it is, increment the consonants counter. c. Check if the current character is not a space. i. If it is, increment the vowels counter. d. Increment the spaces counter if the current character is a space. e. e. Increment the character counter for each character in the string. 7. Check for the start of a word or the end of a word: a. If the current character is the first character of the string and not a space, or if the current character is not a space and the previous character is a space, increment the words counter. 8. Print the final counts for consonants, vowels, digits, spaces, characters, and words. 9. End

PROGRAM:

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

int main()
{
	int consonants = 0, digits = 0, vowels = 0, spaces = 0,
character = 0, words = 0;
	char str[] = "My name is Walter Hartwell White. I live at
308 Negra Arroyo Lane, Albuquerque, New Mexico, 87104. This is my
confession.";
	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] <= 'z')
		{
			if (str[i] != 'a' && str[i] != 'e' && str[i] !=
'i' && str[i] != 'o' && str[i] != 'u')
			{
				consonants++;
			}
			else
			{
				vowels++;
			}
		}
		else if (str[i] == ' ')
		{
			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", consonants, vowels, digits,
spaces, character, words);
}
```

RESULT:

```
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 7$ gcc counter.c
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 7$ ./a.out
Consonants: 51
Vowels: 36
Digits: 8
Spaces: 20
Characters: 121
Words: 21cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 7$
```

Program 2	
PROBLEM STATEMENT :	**Write a Menu driven Program to** **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 (Do not use library functions)
ALGORITHM:	1. Start

2. Initialize Program and Display Menu:
   i. Copy string
   ii. Find string length
   iii. Compare strings
   iv. Reverse string
   v. Concatenate strings
   vi. Convert lowercase to uppercase
   vii. Exit

3. a. Prompt the user to select their desired operation from the menu.
   b. Get the user's input.

4. Using switch :
   i. Case 1: Copy String:
   1. Invoke the copy() function to copy the contents of the second string to the first string using a loop to iterate through each character.
   2. Display the copied string using an appropriate output function.

   ii. Case 2: Find String Length:
   1. Call the stringLength() function to determine the length of the first string by counting the number of characters until the null terminator is encountered.
   2. Display the length of the first string using an appropriate output function.

   iii. Case 3: Compare Strings:
   1. Utilize the compareStrings() function to check if the two strings are equal.
      A. If the strings are equal, display a message indicating their equality using an appropriate output function.
      B. Otherwise, display a message indicating their inequality using an appropriate output function.

   iv. Case 4: Reverse String:
   1. Call the reverseString() function to reverse the characters of the first string using a loop to swap characters from the beginning and end of the string until the middle is reached.
   2. Display the reversed string using an appropriate output function.

   v. Case 5: Concatenate Strings:
   1. Employ the concatenateStrings() function to append the contents of the second string to the first string.

A. Use a loop to iterate through the first string until the null terminator is encountered.
B. Append each character from the second string to the first string.
C. Display the concatenated string using an appropriate output function.

vi. Case 6: Convert Lowercase to Uppercase:

1. 1. Call the convertToUpper() function to convert all lowercase characters in the first string to uppercase.
   A. Use a loop to iterate through the first string.
   B. For each lowercase character, subtract 32 from its ASCII value to convert it to uppercase.
   C. Display the uppercase version of the first string using an appropriate output function.

vii. Case 7: Exit:

1. Exit with return value 0

vii. Case default:

1. Print Invalid choice. Please try again

5. End

PROGRAM:

```c
#include <stdio.h>


void copy(char destStr[], char srcStr[])
{
   int i = 0;
   while (srcStr[i] != '\0')
   {
       destStr[i] = srcStr[i];
       i++;
```

```
    }
    destStr[i] = '\0';
}

int stringLength(char str[])
{
    int length = 0;
    while (str[length] != '\0')
    {
        length++;
    }
    return length;
}

int compareStrings(char str1[], char str2[])
{
    int i = 0;
    while (str1[i] != '\0' && str2[i] != '\0')
    {
        char char1 = str1[i];
        char char2 = str2[i];
        if (char1 >= 'A' && char1 <= 'Z')
        {
            char1 += 32;
        }
        if (char2 >= 'A' && char2 <= 'Z')
        {
            char2 += 32;
        }

        if (char1 != char2)
        {
            return 0;
        }
        i++;
    }
}

void reverseString(char str[])
```

```
{
   int n=0;
   char temp[20];
   n=stringLength(str);
   for(int i=0;i<n;i++)
   {
       if(str[i]!='\0')
       {
           temp[i]=str[n-i-1];
       }
   }
   printf("Reversed String Is :%s\n",temp);
}

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

void convertToUpper(char str[])
{
   int i = 0;
   while (str[i] != '\0')
   {
       char c = str[i];
       if (c >= 'a' && c <= 'z')
       {
           str[i] = c - 32;
```

```
        }
        i++;
    }
}

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

    printf("Enter the first string: ");
    fgets(str1, 100, stdin);
    printf("Enter the second string: ");
    fgets(str2, 100, stdin);
    printf("Menu:\n");
    printf("1. Copy string\n");
    printf("2. Find string length\n");
    printf("3. Compare strings \n");
    printf("4. Reverse string\n");
    printf("5. Concatenate strings\n");
    printf("6. Convert lowercase to uppercase\n");
    printf("7. Exit\n");

    while (1) {
        printf("Enter your choice: ");
        scanf(" %c", &choice);

        switch (choice) {
            case '1':
                copy(str1, str2);
                printf("Copied Result: %s\n", str1);
                break;
            case '2':
                printf("Length of the first string:
%d\n",stringLength(str1));
                break;
            case '3':
                if (compareStrings(str1, str2)) {
                    printf("Strings are equal\n");
```

```
                } else {
                    printf("Strings are not equal\n");
                }
                break;
            case '4':
                reverseString(str1);
                break;
            case '5':
                concatenateStrings(str1, str2);
                printf("Concatenated Result: %s\n", str1);
                break;
            case '6':
                convertToUpper(str1);
                printf("Uppercase Result: %s\n", str1);
                break;
            case '7':
                return 0;
            default:
                printf("Invalid choice. Please try again\n");
        }
    }

    return 0;
}
```

RESULT:

```
Enter the first string: Hello
Enter the second string: World
Menu:
1. Copy string
2. Find string length
3. Compare strings
4. Reverse string
5. Concatenate strings
6. Convert lowercase to uppercase
7. Exit
Enter your choice: 2
Length of the first string: 5
Enter your choice: 3
Strings are not equal
Enter your choice: 4
Reversed String Is :olleH
Enter your choice: 5
Concatenated Result: HelloWorld
Enter your choice: 6
Uppercase Result: HELLOWORLD
Enter your choice: 1
Copied Result: World
Enter your choice: 7
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experimen
```

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. Define the function palindromeornot which takes a string str, a starting index start, and an ending index end. A. It enters a while loop that continues as long as the start index is less than the end index. This loop is responsible for checking the characters at symmetric positions in the string to determine whether it is a palindrome.

B. Inside the while loop, the function compares the characters at indices start and end in the string str using the tolower function to not cause error if the first letter in word is in caps.

C. If the characters at the start and end indices are not equal, the function immediately returns 0, indicating that the string is not a palindrome.

D. If the characters at the symmetric positions are equal, the function increments the start index and decrements the end index to continue checking the next set of symmetric characters.

E. Once the while loop terminates, the function has checked all the necessary character pairs, and if no mismatches were found, it returns 1, indicating that the given substring is a palindrome.

3. In Main function :

I. Declare a character array text to store the input sentence.
II. Prompt the user to input a sentence using printf.
III. Read the input sentence using fgets.
IV. Get the length of the input sentence using strlen.
V. Initialize variables start and end to 0 to keep track of the indices of the current word being processed.
VI. Iterate through the characters in the input sentence using a for loop:
   A. If the current character is a space or newline character, set the variable end to the previous index.
   B. Check if the substring from start to end is a palindrome using the palindromeornot function.
   C. If it is a palindrome, print the palindrome substring using another for loop.
   D. Print a tab character after printing the palindrome to separate multiple palindromes.
   E. Update the start index to the next position after the space.

4. End

PROGRAM:

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

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


int palindromeornot(char str[], int start, int end)
{
   while (start < end)
   {
       if (tolower(str[start]) != tolower(str[end]))
       {
           return 0;//not
       }
       start++;
       end--;
   }
   return 1;
}

int main()
{
   char text[100];
   printf("Enter Sentence: ");
   fgets(text, 100, stdin);
   printf("The palindromes are: ");
   int len = strlen(text);
   int start = 0;
   int end = 0;
   for (int i = 0; i < len; i++)
   {
       if (text[i] == ' ' || text[i] == '\n')
       {
           end = i - 1;
           if (palindromeornot(text, start, end))
           {
               for (int j = start; j <= end; j++)
               {
                   printf("%c", text[j]);
               }
               printf("\t");
```

```
            }
            start = i + 1;
        }
    }
    return 0;
}
```

RESULT:

```
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 7$ gcc palindrome.c
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 7$ ./a.out
Enter Sentence: I AM SURE THE DEED IS ON THE LEVEL MADAM
The palindromes are: I  DEED    LEVEL   MADAM   cyclops@cyclops:~/Desktop/P
```

CONCLUSION:	**I HAVE UNDERSTOOD HOW TO IMPLEMENT STRING.H AND CTYPE.H TO SOLVE PROBLEMS RELATING TO VARIOUS TEXT PROCESSING PROBLEMS**