BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)

CS F111 Computer Programming

LAB SESSION #13

(Strings and File Handling)

Strings

1. Implement a function convertLower() to convert a given string into one that contains entirely lower case letters. The function performs the conversion on the string received as its argument. First, implement your function using ASCII codes only. Next, use the library function tolower() to do the job. Remember to include the header <ctype.h>. You can learn more about this function by reading the manual pages.

[Note: There are a suite of useful library functions like **tolower()** and **toupper()** available in the <ctype.h> library. Get to know them by typing **man isalpha** at the shell prompt. You should be familiar with the first 13 functions – starting from **isalnum()** and ending with **isblank()**, whose descriptions are also provided in the manual page, so you can use them in your programs when needed.]

2. Write a program to accept an English sentence and an English word from the user (using scanf() only), and then to check whether the word exists in the sentence or not. If it occurs, the program should print out how many times the word occurs in the sentence and then remove all the occurrences of the word from the sentence. For taking the input, you have to use input redirection, that is, you have to create a file named input.txt and then take the input from input.txt instead of the keyboard. Eg. **./a.out < input.txt.** Do not use any string processing functions.

A sample input.txt could be as shown below

```
hi hello how are you hello hi
hi
```

input.txt

```
No. of occurrences of hi is 2
output string is: hello how are you hello
```

3. Write a C program that receives two strings S1 and S2 as input. Then it compares S1 and S2, and then copy the two strings in a new string S3 in alphabetical order. If both the strings are equal then either S1 or S2 is copied into S3.

Few sample I/P and O/P instance is given below:

S1 = Programming S2 = Computer
S3 = Computer Programming

S1 = Computer S2 = Computer
S3 = Computer

S1 = Computer S2 = Program
S3 = Computer Program

Note: To achieve the task, you are not allowed to any string processing functions.

4. Write a C program that counts the number of vowels, consonants, digits and other symbols in a given line of text. Read the line of text using (a) gets, and (b) scanf.

5. Write a C program which uses **strstr()** function *repeatedly* and performs the task of locating *the total number of occurrences* of a small string in a huge text. Take input from a file.

Example:
Input Text: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Search string: is
Output: The number of occurrences of "is" are 5

6. Given a file - **numbers.txt**, which contains one integer number per line. The first line of the above file contains the total number of integer numbers in the file. Write a program to read all the numbers from the above file using file operations into a dynamically created array. Then compute their average and display.

7. Given a file – **twoDPoints.csv** that contains two-dimensional points (**(x,y) pairs**). Every line in this file is a **x,y** pair (separated by comma). The first line contains the number of points in the file. You need to store all the points of this file into a 2-D array (say pointsArray). Then for each two-dimensional point stored in the pointsArray, you need to find the closest other point within the same array and store them in another array known as closePointsArray, which is also a 2-D array. Finally, print all the two-dimensional points stored in the closePointsArray.