

# DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

## Assignment Set: 1

### **Problem No: 6**

#### Problem Statement:

Store the names of your classmates according to roll numbers in a text file one name per line. Write a program to find out from the file, the smallest and largest names and their lengths in number of characters. Write a function to sort the names alphabetically and store in a second file.

#### Solution Approach:

Firstly input of students is taken in a text file and stored. Now this file is opened in read mode ("r"), and length of each student name is compared with pre-initialized variables "max" and "min", and the same are updated as required. A counter variable is incremented by 1 in each iteration to count the number of students. This process is ended when we reach EOF.

Next a 2-D array of strings is made and the corresponding names from the file are stored. This array is modified by sorting the names in alphabetical order using Bubble Sort algorithm.

Finally, a new file is created and the array of strings is written into it, followed by the Student names with maximum and minimum lengths respectively.

(NOTE: To find the length of the characters a function is defined which counts the length excluding the spaces)

# DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

## Structured Pseudocode:

FUNCTION LENGTH(CHAR \*NAME):

    C=0

    L=STRLEN(NAME)

    FOR i=0 TO L-1 DO:

        IF(!ISSPACE(NAME[i]))

            C = C + 1

    RETURN C

MAIN():

FILE \*OP = FOPEN("FILE1.TXT", "W")

WHILE(TRUE) DO:

    FGETS(STR, SIZEOF(STR), STDIN)

    FPUTS(STR, OF)

FCLOSE(OP)

FILE\* READ = FOPEN("FILE1.TXT", "R")

WHILE(FGETS(NAME, READ)!=NULL) DO:

    IF(LENGTH(NAME)>MAX)

        MAX=LENGTH(NAME)

    IF(LENGTH(NAME)<MIN)

        MIN=LENGTH(NAME)

    COUNT = COUNT + 1

ARRAY [COUNT][50]

FOR i=0 TO COUNT-1 DO:

    FSCANF(READ, ARRAY[i])

    i = i+1

# DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

FOR i=0 TO COUNT-1 DO:

    FOR j=i+1 TO COUNT-1 DO:

        SWAP (ARRAY[j], ARRAY[j+1])

FILE\* OUT=FOPEN("FLE2.TXT", "W")

FOR i= TO COUNT-1 DO:

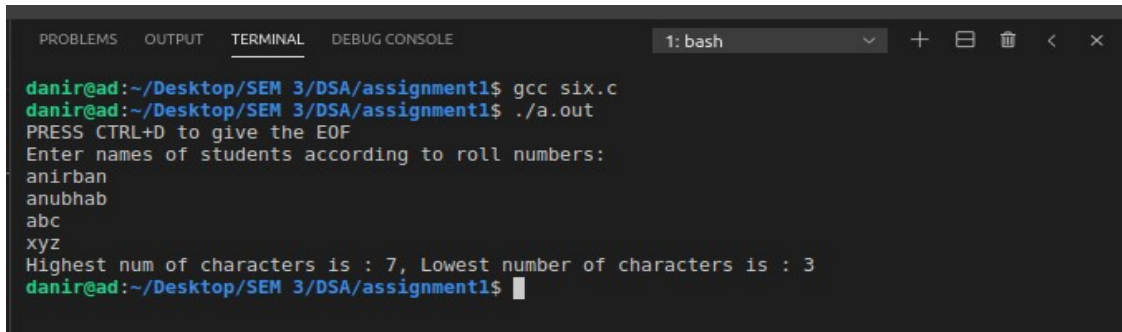
    PRINT(ARRAY[i], OUT)

PRINT(MAX, MIN, OUT)

FCLOSE(OUT)

FCLOSE(READ)

## Results:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ gcc six.c
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ ./a.out
PRESS CTRL+D to give the EOF
Enter names of students according to roll numbers:
anirban
anubhab
abc
xyz
Highest num of characters is : 7, Lowest number of characters is : 3
danir@ad:~/Desktop/SEM 3/DSA/assignment1$
```

## Discussions:

In the worst case scenario the sorting algorithm would take  $n*n$  time to execute, hence the Time Complexity is  $O(n*n)$ . This time can be improved by using more efficient sorting algorithms like quick and merge sort. The space complexity is  $O(n)$ .

# DSA REPORT

**Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21**

## Source Code:

FILE NAME:

Code – “six.c”

Binary Files – “6th\_in.txt” & “6th\_out.txt”

(can be found in the following link: [https://drive.google.com/drive/folders/1-](https://drive.google.com/drive/folders/1-nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing)

[nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing](https://drive.google.com/drive/folders/1-nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing))