

CS19003 Programming and Data Structures Lab

Assignment Set 6

May 9, 2023

INSTRUCTIONS

1. There are two assignments in this Lab. You need to submit each of the three assignments separately. It is advisable to submit each assignment as you complete it, rather than wait for the end to submit everything.

1. [Filename: **set6asg1.c**]

Array of Strings. In this assignment, you are required to handle strings. You may use the standard C library functions on strings, such as `strcat()`, `strcpy()`, and `strcmp()`.

- (a) Write a function, `readinp()`, which reads a set of words from a text file, `input.dat`, and stores them into an array, `A`, of strings. You may assume that there will be at most 100 strings, each of length at most 20. The array, `A`, can be global.
- (b) Write a function, `findinA(char *key)`, that finds and returns the number of occurrences of the `key` in the array, `A`.
- (c) Write a function, `modeofA(char *key)`, that finds a word, `key`, that occurs most number of times in the array `A`. If there are multiple such words, then returning any one of them is okay. The function returns the number of occurrences of this key.

Write a C program that uses the above functions to read the text and a key, to find and report whether that key exists in the text, and to find and print the most frequent word and its frequency.

2. [Filename: **set6asg2.c**]

Two-Dimensional Array of Strings. This assignment follows on the footsteps of the previous assignment. Here you are to read the text into a two-dimensional array, `B[][]`, of strings, such that the element, `B[j][k]`, is the $(k-1)^{\text{th}}$ word of the $(j-1)^{\text{th}}$ line. There will be bonus points for dynamically allocating the array `B`. Then print the text column-wise, that is, print the first word of each row in a line, followed by the second word of each row in the next line, and so on.

Suppose the input text is:

To decipher it the recipient has to work out the
column lengths by dividing the message length by the key
length. Then he can write the message out in columns
again then re-order the columns by reforming the key word.
Columnar transposition continued to be used for serious purposes as
a component of more complex ciphers at least into the
1950s.

Then the output text should be as follows:

To column length again Columnar a 1950s.

decipher lengths Then then transposition component

it by he re-order continued of

the dividing can the to more

recipient the write columns be complex

has message the by used ciphers

to length message reforming for at

work by out the serious least

out the in key purposes into

the key columns word as the