

Assignment VII

Submit all the programs separately against each assignment (i.e. asgn7a, and asgn7b) in the Moodle System 15 minutes before the end of the laboratory session.

All the results for each assignment should be submitted together in a separate file (named result.txt). Provide the result in a separate output file (named, result_<assgn><no>.txt). Use standard output redirection feature to generate the output file.

- (a) Write a function **ConvertWords(.)** which takes an English sentence as an argument and stores the sequence of words present in the sentence as a list of strings in a 2-D array (passed as an argument in the function). The function also returns the number of words. If the sentence does not have any word, it returns 0. Write a program, which reads an English sentence, and converts into a sequence of words in a 2-D array. It forms another sentence by reversing the sequence of words in the list and prints the converted sentence by appending a full stop.

For example,

Given an input string, “**India is an incredible country.**”,

- (i) it stores following sequence of words in a 2-D array
- India**
is
an
incredible
country
- (ii) Then it prints the following sentence
- country incredible an is India.**

Run your program for the following input data set and provide the results with input datasets in a separate output file.

- (i) Where the mind is without fear, and the head is held high!

- (ii) United we stand, united we fall.
- (iii) Jack and Jill went up the hill to fetch a pail of water.

(b) Write a function *encrypt(.)* which takes two strings *s1* and *s2* as arguments, and also *a positive number* as an encryption key. The function takes the characters of the string *s1* one by one from left to right and also reads the digits of the number from right to left and cyclically shift the character following the alphabetic order of the English language by the corresponding digits. If all the digits of the encryption key are exhausted, the process continues by starting from the same positive number to encrypt the successive alphabets in the same way by using its digits from right to left. The encrypted string is returned in *s2*.

For example, given an input string “Lazy” and encryption key “56”, the encrypted string will be “Rffd”.

Write also another function *decrypt(.)*, which takes encrypted string *s1* and *the key (a positive number)* as inputs, and gets back the original string in *s2*, which is also passed as a parameter to the function.

For example, given the input string “Rffd” in *s1* and the key “56”, the function would get the string “Lazy” in *s2*.

Write a main program which reads an English sentence and an encryption key. The program prints its encrypted string by applying the function *encrypt(.)* to each of its words. The program also gets back the original string from the encrypted string by applying the function *decrypt(.)* with *the same key*.

Run your program for the following input data set, and provide the results with input datasets in a separate output file.

- (i) Where the mind is without fear, and the head is held high!
- (ii) United we stand, united we fall.
- (iii) Jack and Jill went up the hill to fetch a pail of water.