

**Department of CSE**

Subject: Data Structures and Algorithms

Assignment 3

Semester: 3rd Sem

Session: August-December 2018

Due date:

**Instructions**

1. This assignment is worth 25 marks of your semester score.
2. Approximate time required to complete this assignment is 8 hours.
3. The Assignment questions are programming based questions on C Language.
4. Format of submission: Soft Copy
5. No Delay in submission of Assignment else will not be accepted.
6. No plagiarism or copying other student’s work. If found, both of the student's assignment will be cancelled.

**ASSIGNMENT-3**

This Assignment contains 5 Programming questions in total.

Objectives: The purpose of the assignment is to make students practise the programming questions so as to improve their technical skills.

Marking Scheme: 5 marks per question (All questions must be answered properly, no irrelevant matter needed)

1. Write a program in C language that can do the following tasks:

* reads a file
* breaks each line into words
* strips whitespace and punctuation from the words
* Converts them to lowercase.

Input: The File is attached below.

Output: As mentioned above.

2. There is an array given with the following elements: 20, 15, 25, 2, 30, 95, 41, 68, 4, 39. The array is an unsorted array. Write a program in C which searches the element 41 and 65 from the array with the total time complexity of O(n).

3. There is an array of 12 elements. The elements are stored from A[2] to A[14] positions or locations in the memory. We need to delete an element from the array at the location A[7]. Write a program which delete the element at the location given and the deletion process must reduce the size of the array so that this chunk of memory can be used by other elements.

4. Consider a link list with 6 elements. After inserting all the elements in the list, the elements are now taken out of the list one by one. The elements when all deleted are arranged in a way that the one which is inserted at first iteration comes at first spot and so on. Apply this first in first out strategy and display the result. The operations must be done in the following order:

* Insert()- insert the 6 elements in the list
* Delete()- delete elements one by one
* Find out the method in data structure which can use this delete() operation and apply a first in first out strategy
* Output will be an empty list.

5. Write a function for a binary Search Tree to check whether the tree is a binary search tree or not. The conditions for the searching are as follows:

* Input: a pointer to the root node of the binary tree and a search key to be located
* Output: a pointer to that node (if found) or return NULL (not found)