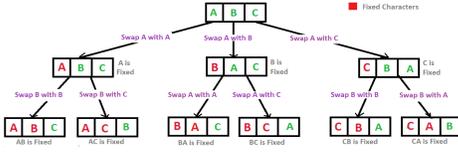


Roll No.	Assignments
CSM16020	 <p data-bbox="925 241 1331 336">Implement an m-way tree that holds all combinations of a given m-length string in the leaf nodes and displays them.</p>
CSM16001	<p data-bbox="451 409 1347 514">Use a priority queue to manage execution of three programs according to their execution time. The programs are added to the queue with their execution time used as the priority. The execution of the simulation proceeds by repeatedly pulling the front of the queue and executing the program thereon.</p>
CSM16029	<p data-bbox="451 525 1347 693">Implement a program that runs a check on each line of text to see if the lines are the same if read from top to bottom or bottom to top. Write a program that reads in several lines of text and through the use of stacks and/or queues decides if the passage reads the same whether you read the lines top-to-bottom or bottom-to-top. For example, the following text reads the same whether you read the lines from top to bottom or bottom to top: (a poem by J.A. Lindon).</p> <p data-bbox="519 697 860 913"> As I was passing near the jail I met a man, but hurried by. His face was ghastly, grimly pale. He had a gun. I wondered why. He had. A gun? I wondered...why, His fact was ghastly! Grimly pale, I met a man but hurried by, As I was passing near the jail. </p> <p data-bbox="451 924 1347 1155">Consider upper- and lower-case versions of a letter to be the same letter. Consider word boundaries to be significant, so for example, the words in the first line must read the same as the words in the last line in order to pass the test (as opposed to just the letters reading the same); however, consider all word delimiters as being equivalent, i.e., a punctuation, or any number of spaces, the beginning or end of a line, or any combination of these are all considered to be equivalent. The end of the passage should be marked by a line containing only the word öendö spelled with any combination of upper- and lower-case letters, and possibly with blanks before and/or after it.</p>
CSM16009	<p data-bbox="451 1186 1347 1260">Implement the telephone directory using an efficient data structure. [You want to store a phone book, which supports looking up a phone number by name, as well as adding and removing entries.]</p>
CSM16005	<p data-bbox="451 1276 1347 1648">Suppose three movies (Bahubali, Moana and Pink) are on display in a cinemax. Three different counters are distributing tickets for each movie. To purchase a particular movie ticket, you are required to stand in the respective movie counter queue. Consider the above scenario and write a program which asks a user to choose the movie for which he/she would like to purchase a Ticket. While choosing the movie ticket, user should be asked to enter his/her social group [F for female, M for male and K for kids]. The program should take input for all the three movies and display the total number of visitors for each movie. Display total revenue of cinemax and total revenue of each movie (Suppose price of one ticket is Rs. 1000/-). Display the best business earning movie of the cinema. Check the popularity of each movie amongst social groups (females, males or kids). Use a 2-3 tree with social group in the last level and no duplicating of node values. (Hint. Use the concept of siblings as taught in class for the last level).</p>
CSM16004	<p data-bbox="451 1665 1347 1816">Parse a file, and put all of the words in a BST. You will use the BST to collect data about the number of times a word was found in the file and print the statistics. You should make no assumptions about which letters are in the middle (like M). The first word you encounter will be the root. If the next word is greater, put it to the right. If it is less, put it to the left. It is possible that the tree you make will be very very sparse (think what happens when the first word is zylberstein).</p>
CSM16003	<p data-bbox="451 1837 1071 1858">Use the selection sort algorithm to implement a priority queue.</p>
CSM16006	<p data-bbox="451 1879 1347 1900">Write a program to implement linked list data structure. In this problem every node</p>

	<p>should have six values, the name of node and a pointer. Your program should prompt the user to enter total number of nodes in the list. After this give the name to node and save 6 different values in node. Process should be repeated for each node. After entering values of all the nodes, sum the values in each node and sort the list in ascending order (on the basis of sum calculated in each node) i.e. node with smallest sum comes first and node with largest sum comes last. Note: Values within a single node should not duplicate. If a user tries to enter any duplicate value WITHIN A NODE then show a warning message and ask user to enter value again.</p>
CSM16021	<p>Write a program that creates a database of student's information for a department. The database should have the following fields: roll no., the first and last names, a course code, and a grade for a student. The program should have the following functions: i) insert a new record (insertion sorted according to last name), ii) delete a record based on roll no. iii) search the database (by last name) and iv) find the class average for a course. To compute the class average, first search the field with the course code. Only average those grades that match the proper course code.</p>
CSM16002	<p>Implement a stack for the following scenario. A waiter washes plates and piles them as a stack. Every plate has an identity number. If the identity number is divisible by 3 the plate breaks, if the identity number is divisible by 5 the plate breaks and also the plate below it and if the identity number is divisible by 7 the plate breaks and also two plates below it.</p>
CSM16008	<p>Write a program that takes a binary tree T and swaps the left and the right children of every node. [reassign pointers not just values]</p>
CSM16016	<p>Implement a Binary Search Tree that will store a sequence of words. The tree will maintain a unique set of words read in from a text file. Case does not matter. For this exercise, there is no requirement to take care of punctuation in the input file. Each node in the tree will keep a tally of how many times the word appears in the file. When a word is read in from the file, add it to the tree. If the word is already in the tree, increment its count. Once the tree is built, the user can search for a word. The output will either report that the word is not in the tree, or if it is, how many occurrences of the word there are. The user can delete an occurrence of a word. If the deletion causes the count to be zero, delete the node with that word from the tree. When the user deletes a word, print the word and its remaining frequency and if it is removed from the tree, print a message stating what word has been deleted from the tree. Provide a way to both print an inorder traversal of the tree and save an inorder traversal to a file.</p>
CSM16014	<p>Generate all prime numbers below an integer value and store them in a linked list in sorted order. Then every alternate node is swapped [pointer reassigning].</p>
CSM16010	<p>Implement a binary search tree where every node consists of the names of students in a class and the Total Ascii value of the name will be the value based on which inserting is to be done.</p>
CSM16011	<p>Implement a data structure to store the stations of a public transportation line. New stations can be added to both ends of the line, but not between existing stations. You should be able to traverse the line in both directions.</p>
CSM16012	<p>Use a data structure to implement a call center. When a client calls, his call should be stored until there is a free operator to pick it up. Calls should be processed in the same order they are received.</p>
CSM16015	<p>Construct a threaded binary tree and convert it into a binary tree.</p>
CSM16013	<p>Write the pseudo-code for a non-recursive function to insert a node into a BST. The elements inserted should be popped from a stack and should be divisible by 2.</p>
CSM16025	<p>Implement a BST and convert it into a sorted circular linked list.</p>
CSM16017	<p>Consider a telephone system that will allow the user to keep track of calls automatically. This telephone system can store the telephone numbers of the 10 most recent callers. Once the limit of 10 is reached, and another call is made, the least recent number is deleted to make room for the most recent number. When the user wishes to review his/her calls, the user can choose between: 1. review new calls 2. review old calls 3. Exit. Use STACK and QUEUE data structure.</p>

CSM16018	Implement a linked list then take in an integer number k, skip k nodes and then reverse k nodes, till the end.
CSM16019	Use a data structure to implement a call center. When a client calls, his call should be stored until there is a free operator to pick it up. Calls should be processed according to some priority of job urgency.
CSM16022	Implement a linked list and delete the first n nodes and form a Stack. (n is user input)
CSM16023	Write a program to swap pairwise elements of a given linked list by changing links
CSM16028	Write a program to check if the linked list is a palindrome.
CSM16026	Implement a program that pops alternate elements from a STACK and builds a new Stack from it. Finally you will have two STACKS: one the original with alternate elements deleted and the other the new stack.
CSM16027	Use two sorted queues and form a BST from them. Deleting from the queues alternately and inserting into the BST.
CSM16030	Write a program to implement a BST by using an array. The values to be inserted are words.
CSM16031	Implement a Stack, split it into two stacks while maintaining the order of insertion.