Bansilal Ramnath Charitable Trust's

Vishwakarma Institute Of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

View Assignment

AssignmentNo	ProblemNo	Concept.	ProblemStatement	Comments	Format	Type
1	1	Sorting & Searching	WAP to implement Bubble sort and Quick Sort on 1D array of Student structure (contains student_name, student_roll_no, total_marks), with key as student_roll_no. And count the number of swap performed.		.txt/.doc	individual
1	2	Sorting & Searching	WAP to implement Insertion sort and Merge Sort on 1D array of Student structure (contains student_name, student_roll_no, total_marks), with key as student_roll_no. And count the number of swap performed.		.txt/.doc	individual
1	3	Sorting & Searching	WAP to implement Selection sort and Bucket Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.		.txt/.doc	individual
1	4	Sorting & Searching	WAP to implement Shell sort and Heap Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.		.txt/.doc	individual
1	5	Sorting & Searching	WAP to implement Insertion sort and Quick Sort on 1D array of Student structure (contains student_name, student_roll_no, total_marks), with key as student_roll_no. And count the number of swap performed.		.txt/.doc	individual
1	6	Sorting & Searching	WAP to implement Selection sort and Merge Sort on 1D array of Student structure (contains student_name, student_roll_no, total_marks), with key as student_roll_no. And count the number of swap performed.		.txt/.doc	individual
1	7	Sorting & Searching	WAP to implement Shell sort and Bucket Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.		.txt/.doc	individual
1	8	Sorting & Searching	WAP to implement Bubble sort and Heap Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.		.txt/.doc	individual
1	9	Sorting & Searching	WAP to implement Bucket Sort and Quick sort on 1D array of Faculty structure (contains faculty_name, faculty_ID, subject_codes, class_names), with key as faculty_ID. And count the number of swap performed		.txt/.doc	individual
1	10	Sorting & Searching	WAP to implement Merge Sort and Heap Sort on 1D array of Faculty structure (contains faculty_name, faculty_ID, subject_codes, class_names), with key as faculty_ID. And count the number of swap performed.		.txt/.doc	individual
1	11	Sorting & Searching	You hav e a fleet of N cars waiting for repair, with the estimated repair times rk for the car Ci , 1 ? k ? N. What is the best repairschedule (order of repairs) to minimize the total lost time for being out-of-service. How much computation is needed to find the lost service-times all schedules?		.txt/.doc	individual
1	12	Sorting & Searching	Write a program to arrange the data of the faculties recruited in the institute. There are three categories of faculties in the every deartment namely professor, Associate professor, and assistant professor. Recruitment is done as mentioned below. 1. Every professor has two associate professors. 2. Every Associate has two assistant professors. Data is given randomly. Suggest suitable sorting method and implement.		.txt/.doc	individual
1	13	Sorting & Searching	Assume that an array A with n elements was sorted in an ascending order, but two of its elements swapped their positions by a mistake while maintaining the array. Write a code to identify the swapped pair of elements and their positions in the asymptotically best possible time. [Assume that all given elements are distinct integers.]		.txt/.doc	individual
1	14	Sorting &	Using Quick sort, assign the roll nos. to the students of your class as per		.txt/.doc	individual

		Searching	their previous years result. i.e. topper will be roll no. 1.			
1	15	Sorting & Searching	Arrange the list of employees as per the average of their height and weight by using appropriate sorting method.		.txt/.doc	individual
1	16	Sorting & Searching	Given a set of points Pi , 1 ? i ? N (? 2) on the x-axis, find Pi and P j such that |Pi ? P j | is minimum. e.g. P1 | P2 | P3 | P4 | P5 | P6 {P4 , P6} is the closest pair.		.txt/.doc	individual
2	1	Searching Techniques	WAP to search a particular students record in 'n' number of students pool by using Binary search with and without recursive function. Assume suitable data for student record.		.txt	individual
2	2	Searching Techniques	WAP to search a particular employees record in 'n' number of employee's pool by using Fibonacci search with and without recursive function. Assume suitable data for employee record.		.txt	individual
2	3	Searching Techniques	WAP to simulate faculty database as a hash table. Search a particular faculty by using MOD as a hash function for linear probing method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	4	Searching Techniques	WAP to simulate faculty database as a hash table. Search a particular faculty by using 'divide' as a hash function for linear probing with chaining without replacement method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	5	Searching Techniques	WAP to simulate faculty database as a hash table. Search a particular faculty by using MOD as a hash function for linear probing with chaining with replacement method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	6	Searching Techniques	WAP to simulate an application of Divide and Conquer methodology to search minimum and maximum number simultaneously from a pool of random numbers entered by the user using recursive function.		.txt	individual
2	7	Searching Techniques	WAP to store k keys into an array of size n at the location computed using a hash function, loc = key % n, where k<=n and k takes values from [1 to m], m>n. To handle the collisions use the following collision resolution techniques, a. Linear probing with chaining b. Rehashing technique.		.txt	individual
2	8	Searching Techniques	WAP to simulate employee database as a hash table. Search a particular faculty by using Mid square method as a hash function for linear probing method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	9	Searching Techniques	WAP to simulate faculty database as a hash table. Search a particular faculty by using 'divide' as a hash function for linear probing with chaining without replacement method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	10	Searching Techniques	WAP to simulate employee database as a hash table. Search a particular faculty by using Mid square Method as a hash function for linear probing with chaining with replacement method of collision handling technique. Assume suitable data for faculty record.		.txt	individual
2	11	Searching Techniques	In a lab there were 10 computers which are having numbers as 21 to 30. Students registered for a lab are 10 only. Design appropriate hash function to assign a computer to every student. Now assume that three students are absent of this batch so another batch students want to use computers in this lab. Allocate the computers to these new students by using linear probing with chaining without replacement.		.txt	individual
2	12	Searching Techniques	In Computer Engg. Dept. of VIT there are S.E., T.E., and B.E. students. Assume that all these students are on ground for a function. We need to identify a student of S.E. div. (X) whose name is "XYZ" and roll no. is "17". Apply appreopriate searching method to identify a required student.		.txt	individual
2	13	Searching Techniques	WAP to store k keys into an array of size n at the location computed using a hash function, loc = key / n, where k<=n and k takes values from [1 to m], m>n. To handle the collisions use the following collision resolution techniques, a. Linear probing with chaining with replacement b. Quadratic probing technique.		.txt	individual

2	14	Searching Techniques	In a lab there were 15 computers which are having numbers as 35 to 50. Students registered for a lab are 14 only. Design appropriate hash function to assign a computer to every student. Now assume that two students are absent of this batch so another batch's three students want to use computers in this lab. Allocate the computers to these new students by using linear probing with replacement.		.txt	individual
2	15	Searching Techniques	Consider two sets of integers, S = {s1, s2, ...,sm} and T = {t1, t2, ..., tn}, m<=n. a.Device an algorithm that uses a hash table of size m to test whether S is a subset of T. b.What is the average running time of your algorithm?		.txt	individual
3	1	Stack	WAP to convert a given Infix expression into its equivalent Postfix expression and evaluate it using stack.		.txt	individual
3	2	Stack	WAP to convert a given Infix expression into its equivalent Prefix expression and evaluate it using stack.		.txt	individual
3	3	Stack	WAP to implement multiple stack i.e. More than two stacks using array and perform following operations on it. A. PUSH, B. POP, C. StackFull D. StackeEmpty E. Display Stack.		.txt	individual
3	4	Stack	WAP to accept a string from user and perform following operations on it using stacks. A. Palindrome B. String Reverse C. String Concat D.String compare		.txt	individual
3	5	Stack	WAP to convert a given Prefix expression into its equivalent Postfix expression and evaluate it using stack.		.txt	individual
3	6	Stack	WAP to convert a given Prefix expression into its equivalent Infix expression and evaluate it using stack.		.txt	individual
3	7	Stack	WAP to implement multiple stack i.e. two stacks using array and perform following operations on it. A. PUSH, B. POP, C. StackFull and D. StackeEmpty E. Display Stack.		.txt	individual
3	8	Stack	WAP to convert a given Postfix expression into its equivalent Prefix expression and evaluate it using stack.		.txt	individual
3	9	Stack	WAP to accept a string of parenthesis and check its validity by using stack.		.txt	individual
3	10	Stack	WAP to convert a given Postfix expression into its equivalent Infix expression and evaluate it using stack.		.txt	individual
3	11	Stack	A person is living in house having 5 rooms. These rooms are adjacent to each other. There is a treasure which is electronically locked and to unlock it we need a code. In last room there is a box in which some decimal number is written. We need to convert that number into binary to open treasure which is kept in room no.1 . We need to move from room no.1 to 2 to 3 and so on and follow the reverse way to come back i.e. from 5 to 4 to 3 etc. Apply suitable logic to implement this scenario by using stacks.		.txt	individual
3	12	Stack	Array is given of size 'n'. We need to divide this array in 'm' numbers, where 2 < m < n. After division each subarray is treated as a stack. If a one stack becomes full we should utilize the space of its next adjacent stack. Write a program to simulate above situation.		.txt	individual
3	13	Stack	WAP to implement following by using stack. A. Factorial of a given number B. Generation of Fibonacci series		.txt	individual
3	14	Stack	Normally we write our name in order of First name, Middle name and Surname. We need to accept these three words in a single string. Write a program to arrange these three words in order of Surname, First name and Middle name using stack only.		.txt	individual
4	1	Queues	We Fly Anywhere Airlines (WFAA) is considering redesigning their ticket counters for airline passengers. They would like to have two separate waiting lines, one for regular customers and one for frequent flyers. Assuming there is only one ticket agent available to serve all passengers, they would like to determine the average waiting time for both types of passengers using various strategies for taking passengers from the waiting lines. WAP to simulate this situation.		.txt	individual
			An operating system assigns job to print queues based on the number of			

4	2	Queues	pages to be printed (1 to 50 pages). You may assume that the system printers are able to print 10 page per minute. Smaller print jobs are printed before larger print jobs and print jobs are processed from a single print queue implemented as a priority queue). The system administrators would like to compare the time required to process a set of print jobs using 1, 2, or 3 system printers. Write a program which simulates processing 100 print jobs of varying lengths using either 1, 2, or 3 printers. Assume that a print request is made every minute and that the number of pages to print varies from 1 to 50 pages. To be fair, you will need to process the same set of print jobs each time you add a printer.The output from your program should indicate the order in which the jobs were received, the order in which they were printed, and the time required to process the set of print jobs. If more than one printer is being used, indicate which printer each job was printed on.		.txt	individual
4	3	Queues	Array is given of size 'n'. We need to divide this array in 'm' numbers. After division each subarray is treated as a queue. If a one queue becomes full we should utilize the space of its next adjecent queue. Write a program to simulate above situation.		.txt	individual
4	4	Queues	Write a menu-driven program that maintains a queue of passengers waiting to see a ticket agent. The program user should be able to insert a new passenger at the rear of the queue, display the passenger at the front of the queue, or remove the passenger at the front of the queue. The program will display the number of passengers left in the queue just before it terminates.		.txt	individual
4	5	Queues	Write a program which simulates the operation of a busy airport which has only two runways to handle all takeoffs and landings. You may assume that each takeoff or landing takes 15 minutes to complete. One runway request is made during each five minute time interval and likelihood of landing request is the same as for takeoff. Priority is given to planes requesting a landing. If a request cannot be honored it is added to a takeoff or landing queue. Your program should simulate 120 minutes of activity at the airport. Each request for runway clearance should be time-stamped and added to the appropriate queue. The output from your program should include the final queue contents, the number of take offs completed, the number of landings completed, and the average number of minutes spent in each queue.		.txt	individual
4	6	Queues	Write a Program to implement double ended queue where user can add and remove the elements from both front and rear of the queue		.txt	individual
4	7	Queues	Write a Program to implement circular double ended queue where user can add and remove the elements from both front and rear of the queue		.txt	individual
4	8	Queues	Write a Program to keep track of patients as they check into a medical clinic, assigning patients to doctors on a first-come, first-served basis.		.txt	individual
4	9	Queues	There is a lift in our college. Students, staff and guest are utilizing it. Students are using the lift when no staff is there. Whenever staff is in the lift students are moving out of the lift. Whenever HOD's / Dean's / Director want to use lift then staff are moving out of the lift. It means every entity which is using the lift is having some priority. High priority entity will be served first. Simulate this situation by using appropriate queue.		.txt	individual
4	10	Queues	Assume that there are three jobs to be done (J1, J2, J3) by using queue. Each is requiring different time for processing i.e. (t1, t2, t3), which is greater than fixed time quantum 'n'. After 'n' time the current job is forcefully preempted/ stopped and remaining task of the current job is added at last of the queue. then next job is taken for processing.		.txt	individual
4	11	Queues	Write a Program to simulate the following situation. Computer is a multitasking device. We need to download some document as well as listen music and play game simultaneously. There is a system queue which decides which task to be done first. Assume that for download application priority is highest and game playing is having lowest priority. After completion of one type of tasks like all download operations then the second queue will be processed.		.txt	individual

4	12	Queues	Write an algorithm Replace that takes a queue and two item. If the first item is in the queue, replace it with the second item, leaving the rest of the queue unchanged.		.txt	individual
4	13	Queues	Write a Program to implement multiple queue i.e. two queues using array and perform following operations on it. A. Addq, B. Delq, C. Display Queue.		.txt	individual
4	14	Queues	There are 'n' number of soldiers surrounded by enemy. They are having only one horse. They decided that one of them will go out and seek the help from outside while others will hold on the post. To decide who will go out to seek help they played a game. they stood in circular way and select some random number 'm' which is smaller than 'n'. then they selected a random person 'x' from which they start counting 'm'. after 'm' number the person who is at that place will be out of the game i.e. he has to wait at post only. Again they started the count 'm' from the next person. this process will go on till one person remains. the last person is the one who will go out with horse and seek the help from outside. Write a program to simulate this situation.		.txt	individual
4	15	Queues	Write a Program for A bank simulation of its teller operation to see how waiting times would be affected by adding another teller.		.txt	individual
4	16	Queues	Array is given of size 'n'. We need to divide this array in 'm' numbers. After division each sub-array is treated as a queue. If a one queue becomes full we should utilize the space of its next adjacent queue. Write a program to simulate above situation.		.txt	individual
4	17	Queues	There are 'n' number of soldiers surrounded by enemy. They are having only one horse. They decided that one of them will go out and seek the help from outside while others will hold on the post. To decide who will go out to seek help they played a game. they stood in circular way and select some random number 'm' which is smaller than 'n'. then they selected a random person 'x' from which they start counting 'm'. after 'm' number the person who is at that place will be out of the game i.e. he has to wait at post only. Again they started the count 'm' from the next person. this process will go on till one person remains. the last person is the one who will go out with horse and seek the help from outside. Wirte a program to simulate this situation.		.txt	individual
5	1	Linked Lists	Given a list, split it into two sublists — one for the front half, and one for the back half. If the number of elements is odd, the extra element should go in the front list. So FrontBackSplit() on the list {2, 3, 5, 7, 11} should yield the two lists {2, 3, 5} and {7, 11}. Getting this right for all the cases is harder than it looks. You should check your solution against a few cases (length = 2, length = 3, length=4) to make sure that the list gets split correctly near the short-list boundary conditions. If it works right for length=4, it probably works right for length=1000. You will probably need special case code to deal with the (length <2) cases.		.txt	individual
5	2	Linked Lists	WAP to perform addition o f two polynomials using singly linked list.		.txt	individual
5	3	Linked Lists	Write an iterative Reverse() function that reverses a list by rearranging all the .next pointers and the head pointer. Ideally, Reverse() should only need to make one pass of the list.		.txt	individual
5	4	Linked Lists	Write an Append() function that takes two lists, 'a' and 'b', appends 'b' onto the end of 'a', and then sets 'b' to NULL (since it is now trailing off the end of 'a').		.txt	individual
5	5	Linked Lists	WAP to perform Multiplication o f two polynomials using singly linked list.		.txt	individual
5	6	Linked Lists	Consider a CopyList() function that takes a list and returns a complete copy of that list. One pointer can iterate over the original list in the usual way. Two other pointers can keep track of the new list: one head pointer, and one tail pointer which always points to the last node in the new list.		.txt	individual
5	7	Linked Lists	WAP to store at most 10 digit integer in a Singly linked list and perform arithmetic operations on it.		.txt	individual
		Linked	WAP to create doubly linked list and perform following operations on it.			

5	8	Lists	A) Insert (all cases) 2. Delete (all cases).		.txt	individual
5	9	Linked Lists	WAP to store at most 10 digit integer in a Doubly linked list and perform arithmetic operations on it.		.txt	individual
5	10	Linked Lists	WAP to merge two sorted Doubly linked lists and display ther result.		.txt	individual
5	11	Linked Lists	WAP to append one doubly linked list to a) the start of the Second list b) the end of the second list.		.txt	individual
5	12	Linked Lists	Implement Push and POP operations of STACK on Doubly linked lists		.txt	individual
5	13	Linked Lists	Implement ADD and DELETE operations of QUEUE on Doubly linked lists		.txt	individual
5	14	Linked Lists	Implement Insertion sort using Singly Linked List		.txt	individual
5	15	Linked Lists	Implement Bubble sort using Doubly Linked List		.txt	individual
5	16	Linked Lists	Write a RemoveDuplicates() function which takes a list sorted in increasing order and deletes any duplicate nodes from the list. Ideally, the list should only be traversed once.		.txt	individual
5	17	Linked Lists	Write a SortedInsert() function which given a list that is sorted in increasing order, and a single node, inserts the node into the correct sorted position in the list. While Push() allocates a new node to add to the list, SortedInsert() takes an existing node, and just rearranges pointers to insert it into the list.		.txt	individual
6	1	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform COPY operation on it.		.txt	individual
6	2	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform EQUIVALENCE operation on it.		.txt	individual
6	3	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Find DEPTH of GLL.		.txt	individual
6	4	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform DELETE operation on it..		.txt	individual
6	5	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Equivalence operation on it.		.txt	individual
6	6	GLL	WAP to create GLL to store multi variable polynomial. Perform COPY operation on it.		.txt	individual
6	7	GLL	WAP to create GLL to store multi variable polynomial. Perform DEPTH operation on it.		.txt	individual
6	8	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Addition operation on it.		.txt	individual
6	9	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Addition operation on it.		.txt	individual
6	10	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform COPY operation on it.		.txt	individual
6	11	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform EQUIVALENCE operation on it.		.txt	individual
6	12	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Find DEPTH of GLL.		.txt	individual
6	13	GLL	WAP to Create A GLL of type A=(a,b, (c,d),e, (f,g)….). Perform DELETE operation on it..		.txt	individual
6	14	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Equivalence operation on it.		.txt	individual
6	15	GLL	WAP to create GLL to store multi variable polynomial. Perform COPY operation on it.		.txt	individual
6	16	GLL	WAP to create GLL to store multi variable polynomial. Perform DEPTH operation on it.		.txt	individual

6	17	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Addition operation on it.		.txt	individual
6	18	GLL	WAP to create GLL to store two different multi variable polynomials. Perform Addition operation on it.		.txt	individual
7	1	TREES	Write a Program to create a Binary Tree and perform following nonrecursive operations on it. a. Preorder Traversal b. Postorder Traversal c. Count total no. of nodes d. Display height of a tree.		.txt	individual
7	2	TREES	Write a Program to create a Binary Tree and perform following nonrecursive operations on it. a. Levelwise display b. Mirror image c. Display height of a tree.		.txt	individual
7	3	TREES	Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Delete • Find • Show		.txt	individual
7	4	TREES	Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Mirror Image • Find • Post order (nonrecursive)		.txt	individual
7	5	TREES	Write a Program to create a Binary Tree and perform following Nonrecursive operations on it. a. Inorder Traversal b. Preorder Traversal c. Display Number of Leaf Nodes d. Mirror Image		.txt	individual
7	6	TREES	Write a Program to create a Binary Tree and perform following Nonrecursive operations on it. a. Inorder Traversal b. Preorder Traversal c. Display Height of a tree d. Find Maximum		.txt	individual
7	7	TREES	You have to maintain information for a shop owner. For each of the products sold in his/hers shop the following information is kept: a unique code, a name, a price, amount in stock, date received, expiration date. For keeping track of its stock, the clerk would use a computer program based on a search tree data structure. Write a program to help this person, by implementing the following operations: • Insert an item with all its associated data. • Find an item by its code, and support updating of the item found. • List valid items in lexicographic order of their names.		.txt	individual
7	8	TREES	You have to maintain information for a shop owner. For each of the products sold in his/hers shop the following information is kept: a unique code, a name, a price, amount in stock, date received, expiration date. For keeping track of its stock, the clerk would use a computer program based on a search tree data structure. Write a program to help this person, by implementing the following operations: • Insert an item with all its associated data. • Find an item by its code, and support updating of the item found. • List valid items in lexicographic order of their names.		.txt	individual
7	9	TREES	Write a Program to create a Binary Search Tree and perform following nonrecursive operations on it. a. Preorder Traversal b. Inorder Traversal c. Display Number of Leaf Nodes d. Mirror Image		.txt	individual
7	10	TREES	Write a Program to create a Binary Search Tree and perform following nonrecursive operations on it. a. Preorder Traversal b. Postorder Traversal c. Display total Number of Nodes d. Display Leaf nodes.		.txt	individual
7	11	TREES	Write a Program to create a Binary Search Tree and perform deletion of a node from it. Also display the tree in nonrecursive postorder way.		.txt	individual
7	12	TREES	Write a Program to create a Binary Search Tree and display it levelwise. Also perform deletion of a node from it.		.txt	individual
7	13	TREES	Write a Program to create a Binary Search Tree and display its mirror image with and without disturbing the original tree. Also display height of a tree using nonrecursion.		.txt	individual
7	14	TREES	You have to maintain information for a shop owner. For each of the products sold in his/hers shop the following information is kept: a unique code, a name, a price, amount in stock, date received, expiration date. For keeping track of its stock, the clerk would use a computer program based on a search tree data structure. Write a program to help this person, by implementing the following operations: • Insert an item with all its associated data. • List expired items in Prefix order of their		.txt	individual

			names. • List all items. • Delete an item given by its code. • Delete all expired items.			
7	15	TREES	Write a program, using trees, to assign the roll nos. to the students of your class as per their previous years result. i.e topper will be roll no. 1.		.txt	individual
7	16	TREES	Write a program to efficiently search a perticular employee record by using Tree data structure. Also sort the data on emp-id in ascending order.		.txt	individual
8	1	Advance Trees	Write a Program to create Inorder Threaded Binary Tree and Traverse it in Inorder and Preorder way.		.txt	individual
8	2	Advance Trees	Write a Program to create Inorder Threaded Binary Tree and Traverse it in Inorder and Postorder way.		.txt	individual
8	3	Advance Trees	Write a Program to implement AVL tree and perform different rotations on it.		.txt	individual
8	4	Advance Trees	Write a Program to create Inorder Threaded Binary Tree and Traverse it in Postorder and Preorder way.		.txt	individual
9	1	Graphs	Write a Program to accept a graph from user and represent it with Adjacency Matrix and perform BFS and DFS traversals on it.		.txt	individual
9	2	Graphs	Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	3	Graphs	Write a Program to implement Kruskal's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	4	Graphs	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	5	Graphs	Write a Program to accept a graph from user and represent it with Adjacency List and perform BFS and DFS traversals on it.		.txt	individual
9	6	Graphs	Write a Program to implement Kruskal's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency Matrix to represent a graph.		.txt	individual
9	7	Graphs	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency Matrix to represent a graph.		.txt	individual
9	8	Graphs	Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	9	Graphs	Write a Program to implement Kruskal's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	10	Graphs	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.		.txt	individual
9	11	Graphs	Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency Matrix to represent a graph.		.txt	individual