

B. P. PODDAR INSTITUTE OF MANAGEMENT & TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING ACADEMIC YEAR: 2018-2019 [ODD SEMESTER] LIST OF EXPERIMENTS

Course: Data Structure & Algorithm Lab Code: CS392 Branch: CSE

Implementation of array operationsI. Static 1-D array representationCOI, POI, CO2POI, CO2array operations2. Insert and delete an element from a specified location in a 1-D static array along with the mother element in a 1-D static array delete duplicates element along with the mother element in a 1-D static array 4. Find the largest and smallest element present in a 1-D static array 5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.PO9, PO10, <b< th=""><th>TOPIC</th><th>LIST OF EXPERIMENTS</th><th>СО</th><th>PO/</th></b<>	TOPIC	LIST OF EXPERIMENTS	СО	PO/
Implementation of array operationsI. Static 1-D array representationCO1, PO1, PO2, PO3, PO4,<				PSO
array operations2. Insert and delete an element from a specified location in a 1-D static arrayCO2PO2,3. Reverse the elements present in a 1-D static array delete duplicates element along with the mother element in a 1-D static arrayPO3,4. Find the largest and smallest element present in a 1-D static array 5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.PO3,5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.PO1,901,PO1,902,PO3,903,PO1,904,PO1,905,PO1,904,PO1,905,PO1,905,PO1,901,PO1,902,PO3,903,PO4,904,PO1,904,PO1,905,PO2,905,PO1,905,PO2,905,PO3,905,PO3,905,PO3,905,PO3,905,PO3,905,PO3,905,PO3,905,PO3,905,PO3,905,PO4,905,PO4,905,PO4,905,PO4,905,PO4,905,PO4,906,PO1,907,PO1,906,PO1,907,PO1,907,PO1,907,PO3,908,PO4,909,	Implementation of	1. Static 1-D array representation	CO1,	PO1,
3. Reverse the elements present in a 1-D static array delete duplicates element along with the mother element in a 1-D static array 4. Find the largest and smallest element present in a 1-D static array 5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.PO3, PO4, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO10, PO2, PS01, PS02CO2 PO1, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO3, PO3, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO3, PO3, PO3, PO3, PO4, PO3	array operations	2. Insert and delete an element from a specified location in a 1-D static array	CO2	PO2,
along with the mother element in a 1-D static arrayPO4,4. Find the largest and smallest element present in a 1-D static arrayPO8,5. Write a program to store the elements in a 2D array and display it andPO9,represent it in row major order & display it.PO10,PO12,PS01,PS02PO12,Stack and Queues:1. Implement a stack using array implementation including the function toCO2Adding & deletingelementsPO4,2. Implement a stack using Linked list.PO4,3. Reverse a string using stack.PO4,4. Representation of Queue and implementation of basic queue operationsPO12,PS01,PS02,PS01,PS03,PO4,PO4,2. Implement a stack using Linked list.PO4,3. Reverse a string using stack.PO9,4. Representation of Queue and implementation of basic queue operationsPO12,PS01,PS01,PS02PO12,Circular Queue1. Implementation of Circular queue using array.CO2Adding & deleting1. Implementation of Circular queue using array.CO2Adding & deleting1. Implementation of Circular queue using array.PO4,PO		3. Reverse the elements present in a 1-D static array delete duplicates element		РОЗ,
4. Find the largest and smallest element present in a 1-D static array PO8,5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.PO10,P012,PS01,PS01PS02PS01PS01,PS02PO10,Stack and Queues:1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsPO3,1. Implement a stack using Linked list. 3. Reverse a string using stack.PO4,4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursionPO10, PO12, PS01, PS01, PS01, PS02,Circular Queue Adding & deleting elements1. Implementation of Circular queue using array.CO2 PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO3, PO3, PO4, PO3, PO3, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3,		along with the mother element in a 1-D static array		PO4,
5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.P09, P010, P012, PS01BookP01, P012, PS02Stack and Queues: Adding & deleting the elements1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack,display the stack elementsCO2, P00, P02, P03, P03, P04, P04, P04, P05, P05, P05, P06, <br< td=""><td></td><td>4. Find the largest and smallest element present in a 1-D static array</td><td></td><td>PO8,</td></br<>		4. Find the largest and smallest element present in a 1-D static array		PO8,
represent it in row major order & display it.P010, P012, PS01, PS01Stack and Queues:1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsCO2P01, PO2, PO2, PO3, PO3, PO3, PO4, PO4, 2. Implement a stack using Linked list. 3. Reverse a string using stack.P08, PO8, PO9, PO9, 4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursionP012, PS01, PS01, PS02, PS03, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS04, PS05, PS04, PS04, PS04, PS04, PS04, PS04, PS05, PS05, PS05, PS05, PS06, <b< td=""><td></td><td>5. Write a program to store the elements in a 2D array and display it and</td><td></td><td>PO9,</td></b<>		5. Write a program to store the elements in a 2D array and display it and		PO9,
Po12, PS01, PS02Stack and Queues:1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsCO2PO1, PO2, PO2, PO3, PO3, PO4, PO4, PO5, Adding & deleting elementsCO2PO1, PO2, PO3, PO3, PO4, PO3, PO9, PO9, A. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursionPO1, PO1, PS01, PS02Circular Queue Adding & deleting elements1. Implementation of Circular queue using array.CO2PO1, PO1, PS02, PS01, PS01, PS02, PS03, PO3, PO4, PO3, PO3, PO4, PO3, PO3, PO4, PO3, PO3, PO4, PO3, PO3, PO4, PO3, PO3, PO4, PO5, PO3, PO4, PO5, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO3, PO4, PO5, PO3, PO4, PO5, PO3, PO4, PO5, PO3, PO4, PO5, PO3, PO4, PO5, PO		represent it in row major order & display it.		PO10,
Image: ProblemProblemProblemStack and Queues:1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsPO2, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO4, PO4, PO4, PO4, PO3, Reverse a string using stack. A Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. S. Solution to Tower of Hanoi using recursionPO2, PO3, PO1, PS01, PS01, PS02Circular Adding & deleting elements1. Implementation of Circular queue using array.CO2 PO1, PO3, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO4, PO3, <b< td=""><td></td><td></td><td></td><td>PO12,</td></b<>				PO12,
Image: space s				PSO1,
1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsPO2, PO3, PO3, PO4, PO				PSO2
Stack and Queues:check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elementsPO2, PO3, PO3, PO4, PO4, PO4, PO4, PO4, PO4, PO4, PO4, PO4, PO4, PO9, PO10, enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursion Adding & deleting elementsPO2, PO3, PO4, PO9, PO10, PO10, PO12, PS01 PS01 PS02Circular Queue Adding & deleting elements1. Implementation of Circular queue using array.CO2 PO1, PO2, PO3, PO4, PO2, PO3, PO4, PO2, PO3, PO2, PO3, PO1, PS01, PO3, PO2, PO3, PO3, PO4, PO3, PO3, PO2, PO3, PO3, PO3, PO3, PO4, PO3, <b< td=""><td></td><td>1. Implement a stack using array implementation including the function to</td><td>CO2</td><td>PO1,</td></b<>		1. Implement a stack using array implementation including the function to	CO2	PO1,
Adding & deleting the elementselement from a stack. Return the top element from the stack,display the stack elementsPO3, PO4, PO4, PO8, PO9, A. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursionPO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO10, PO2, PO2, PO10, PO2, PO3, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3, PO	Stack and Queues:	check whether the stack is empty. Push an element into the stack; pop an		PO2,
the elements elements PO4, 2. Implement a stack using Linked list. PO8, 3. Reverse a string using stack. PO9, 4. Representation of queue and implementation of basic queue operations PO10, enqueue, dequeue, overflow, underflow using linked list. PO12, 5. Solution to Tower of Hanoi using recursion PS01, PS02 PC1 Circular Queue 1. Implementation of Circular queue using array. CO2 Adding & deleting PO4, elements PO3, PO4, PO3, PO4, PO4, PO3, PO3, PO4, PO3, PO4, PO3, PO4, PO4,	Adding & deleting	element from a stack. Return the top element from the stack, display the stack		РОЗ,
2. Implement a stack using Linked list.P08,3. Reverse a string using stack.P09,4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list.P012,5. Solution to Tower of Hanoi using recursionPS01,PS02PS02Circular Queue1. Implementation of Circular queue using array.CO2Adding & deleting elementsP04,P04,P04,P04,P04,P05,P01,P01,P01,P02,P01,P03,P04,P04,P04,P04,P04,P05,P01,P05,P01,P01,P01,P05,P01,P05,P01,P06,P01,P07,P01,P07,P01,P08,P09,P09,P01,P01,P01,P09,P01, </td <td>the elements</td> <td>elements</td> <td></td> <td>PO4,</td>	the elements	elements		PO4,
3. Reverse a string using stack.PO9,4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list.PO10,5. Solution to Tower of Hanoi using recursionPS01,PS02PS02Circular Queue1. Implementation of Circular queue using array.CO2Adding & deleting elementsPO3,PO4,PO3,PO4,PO4,PO4,PO3,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO4,PO5,PO10,PO5,PO10,PO5,PO10,PO5, <td></td> <td>2. Implement a stack using Linked list.</td> <td></td> <td>PO8,</td>		2. Implement a stack using Linked list.		PO8,
4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list.PO10, PO12, PS015. Solution to Tower of Hanoi using recursionPO10, PS01Circular Queue1. Implementation of Circular queue using array.CO2Adding & deleting elementsPO4, PO4, PO4, PO4, PO4, PO4, PO4, PO4, PO5, P		3. Reverse a string using stack.		PO9,
enqueue, dequeue, overflow, underflow using linked list.PO12, PS01, PS025. Solution to Tower of Hanoi using recursionPS01, PS02Circular Queue1. Implementation of Circular queue using array.CO2Adding & deleting elementsPO3, PO4, PO3, PO4, PO3, PO4, PO3, PO3, PO4, PO3,<		4. Representation of queue and implementation of basic queue operations		PO10,
5. Solution to Tower of Hanoi using recursionPSO1, PSO2Circular Queue1. Implementation of Circular queue using array.CO2PO1, PO2, PO3, PO4, PO4, PO8, PO9, PO10, PO10, PO12, PSO1Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedCO1, PO1,		enqueue, dequeue, overflow, underflow using linked list.		PO12,
CircularQueue1. Implementation of Circular queue using array.CO2PO1,Adding & deleting elements1. Implementation of Circular queue using array.PO2,PO3,PO4,PO4,PO4,PO9,PO1,PO9,PO1,PO1,PO1,PO1,PO1,PO1,PO1,PO1,PO1,PO12,PS01,PS01PS01,PS01PS01,PS01PS01,PS02PO1,Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedCO1,		5. Solution to Tower of Hanoi using recursion		PSO1,
CircularQueue1. Implementation of Circular queue using array.CO2PO1,Adding & deleting elementsPO3,PO4,PO3,PO4,PO4,PO9,PO1,PO1,PO9,PO1, <th></th> <th></th> <th></th> <th>PSO2</th>				PSO2
Adding & deleting elementsPO2, PO3, PO4, PO4, PO8, PO9, PO10, PO10, PO10, PO12, PS01, PS01Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedCO1, PO1, PO1, PO1, PS01	Circular Queue	1. Implementation of Circular queue using array.	CO2	PO1,
elementsPO3,PO4,PO4,PO5,PO5,PO5,PO5,PO5,PO5,PO5,PO10,PO	Adding & deleting			PO2,
Merging Problem 1. Merging two sorted arrays so that the resultant array is also sorted PO4, PO4, PO4, PO4, PO4, PO3, PO4, PO4, PO3, PO4, PO4,	elements			PO3,
Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedPO8, PO9, PO10, PO12, PS01, PS02Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedCO1,				PO4,
Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedPO9, PO10, PO10, PO10, PS01, PS02				PO8,
Merging Problem 1. Merging two sorted arrays so that the resultant array is also sorted CO1, PO10, PO12, PS01, PS02 PO12, PS01, PS02 PO12, PS01 PS02				PO9,
Merging Problem 1. Merging two sorted arrays so that the resultant array is also sorted CO1,				PO10,
Merging Problem 1. Merging two sorted arrays so that the resultant array is also sorted CO1,				PU12,
Merging Problem1. Merging two sorted arrays so that the resultant array is also sortedCO1,PO1,				PSO1,
Merging Problem 1. Merging two sorted arrays so that the resultant array is also sorted CO1, PO1,	Manaina Duablan	1. Manaina tana aantad amaana aa that tha manifest anna is alaa aanta l	CO1	PO1
2 Marging two singly linked lists	Merging Problem	2. Marging two sorted arrays so that the resultant array is also sorted	CO1,	PO1,
2. Weiging two singly inited lists $CO2$, $PO2$, $CO4$, $PO2$		2. Merging two singly miked lists	CO_2 ,	PO2,

			PO4,
			PO8,
			PO9,
			PO10,
			PO12,
			PSO1,
			PSO2
Evaluation of	1. Evaluation of postfix expression.	CO2	PO1,
Expressions	2. Conversion from infix to postfix		PO2,
_			PO3,
			PO4,
			PO8,
			PO9,
			PO10,
			PO12.
			PSO1.
			PSO2
Implementation of	3. Representation of single linked lists (create and display it) and then	CO1	PO1.
linked lists:	implementation of relevant operations – add delete from beginning end and at	CO^2	PO2
inserting, deleting	and after specified locations	CO4.	PO3.
inverting a linked list	4 Reversing the links of a singly linked list	CO5	PO4
inverting a minea list	5 Sorting the nodes of a singly linked list in ascending order	000	PO8
	5. Solding the nodes of a singly mixed list in ascending order		PO9
			PO10
			PO12
			PSO1
			PSO2
Implementation of	1 Implementation of stack using linked lists	CO1	PO1
stacks & queues	2 Implementation of queue using linked lists	CO^2	PO2
using linked lists	2. Implementation of queue using mixed lists	CO2,	PO3
using mixed lists		005	PO4
			104, DO9
			PO0,
			PO9,
			PO10,
			PO12,
			PSO1,
		001	PS02
Polynomial addition,	1. Inked representation of polynomials		PO1,
polynomial	2. Addition of two polynomials	CO2,	PO2,
multiplication		005	PO3,
			PO4,
			PO8,
			PO9,

			PO10, PO12, PSO1,
Sparse matrices: addition, multiplication	 Write a program to test a given matrix is sparse or not. If it is sparse then represent it as 3-tuple format. Find the transpose of a sparse matrix 	CO1, CO2	PSO2 PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO10, PO12, PSO1, PSO2
Recursive & Non recursive traversal of trees	 Write a program to create recursive binary search tree Write a program to implement Recursive and non-recursive traversal of a binary search tree - post order, pre-order and in-order traversal 	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Threaded binary tree traversal	Threaded binary tree implementation.	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
AVL tree implementation	Implementation of Height Balanced (AVL) tree	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1,

			PSO2
Hash table	1.Hash tables implementation - concepts search, insert and delete	CO2,	PO1,
implementation:		CO5	PO2,
searching, inserting,			PO3,
deleting			PO4,
			PO8,
			PO9,
			PO10,
			PO12,
			PSO1,
			PSO2
Searching and	1. Searching - Linear/Sequential, Binary (using function)	CO4	PO1,
sorting technique	2. Implementation of bubble sort & modified bubble sort		PO2,
	3. Implementation of insertion sort, selection sort		РОЗ,
	4. Implementation of Quick sort, Merge sort		PO4,
			PO8,
			PO9,
			PO10,
			PO12,
			PSO1,
			PSO2