

**BABU MADHAV INSTITUTE OF INFORMATION TECHNOLOGY, UTU**  
**Integrated M.Sc.(IT) / B.Sc.(IT)**

---

<b>Practical List: 02</b>	
1.	Write a C program to initialize string array say “Language” which contains name of different programming language. Also display all the elements of array using traversing algorithm.
2.	Write a C++ program to initialize/input array named percentage which contains percentage of students from standard 1 to 12. Display percentage in ascending order along with their standard. [Hint: Take the use of Sorting algorithm]
3.	Consider that students are approaching for participating in Volley Ball sports. Only 12 player is allowed to register. Write a C/C++ menu driven program to create an array of students roll number who want to participate in Volley Ball. Based on user choice insert roll number in the array using insertion algorithm. Allow student to de-register (remove) from the team if already registered. Also search and display the position of inputted roll number. Display all registered student’s roll numbers on a screen. Use appropriate message to convey following. No more registration allowed (if all 12 registration done) Still N students can register (if less than 12 registration done) Student is not registered (roll number not found while searching) Can’t remove! Student is not registered (roll number not found while searching) Yet none of the students have registered. Students de-registered from the game successfully. [Note: Perform all these using algorithm]
4.	Write a menu driven program to implement stack of integer numbers-using array. Following operation should be performed: A. Insert element (Push) B. Remove element (Pop) C. Display stack
5.	Consider the stack of Book type (book title, author, publication, price and pages). Implement push, pop and display functionality on book.
6.	Write a program in C++ for the conversion of Infix expression to Postfix expression using a stack.
7.	Write a program in C++ for evaluating the Postfix expression using a stack.
8.	Write a C++ program to find the factorial of an integer ‘n’ with a Stack.
9.	Write a menu driven program to implement problem statement 3 functionality with queue using array: A. Insert/Register Player (Enqueue) B. Remove Player (Dequeue) C. Display the queue
10.	Write a menu driven program to implement problem statement 3 functionality with Deque using array: A. Insert element at Front B. Insert element at Rear C. Remove element at Front D. Remove element at Rear E. Display the queue

**BABU MADHAV INSTITUTE OF INFORMATION TECHNOLOGY, UTU**  
**Integrated M.Sc.(IT) / B.Sc.(IT)**

---

<b>Objective(s)</b>	To vibrant the concept of Array, Stack and Queue.
<b>Pre-requisites</b>	Basic of Object-Oriented Concepts.
<b>Duration for Completion</b>	10 Hours
<b>PSO(s) to be achieved</b>	PSO3: Student will be able to devise and conduct experiments and provide well informed conclusions using recent tools, technologies and industrial trends.
<b>PO(s) to be achieved</b>	PO2: Problem Analysis and Solution: Identify, Analyse and provide the solution for emerging real-world problems with the help of theoretical and practical understanding of tools and technologies.
<b>CO(s) to be achieved</b>	CO1: Identify essential Data Structures and analyse the complexity of algorithms and identify the optimized algorithm. CO2: Recognize problem properties where arrays, stacks, queues, and dequeue are appropriate data structures.
<b>Solution must contain</b>	Source Code with comments and Output Screen Shot
<b>Nature of submission</b>	Handwritten on A4 size blank papers
<b>References for solving the problem</b>	Textbook: Classic Data Structures, Debasis Samanta, PHI.
<b>Post Laboratory questions</b>	<p>1. Convert the following Infix expressions into Prefix and Postfix expression:</p> <p>1. <math>(A + B) * C - D / (E * F)</math>      2. <math>(A + B) / C - (E + F)</math>  3. <math>((A / B) * C) - E</math>      4. <math>(A + B) * (C + D) / (E * F)</math>  5. <math>(A + B - C) * E - F</math></p> <p>2. Translate the following Infix expressions into its equivalent Postfixexpression [in a tabular form]:</p> <p>a) <math>(A + B) * C - D / (E * F)</math>  b) <math>(A + B) / C - (E + F)</math>  c) <math>((A / B) * C) - E</math>  d) <math>(A + B) * (C + D) / (E * F)</math>  e) <math>(A + B - C) * E - F</math></p> <p>3. Evaluate the following postfix expression using a stack [in a tabular form]:</p> <p>a. 15, 3, 2, +, /, 7, +, 2, *  b. 10, 3, *, 7, 1, -, *, 23, +  c. 12, 7, 3, -, /, 2, 1, 5, +, *, +  d. 7, 2, -, 1, 14, -, 1, 2, *, +, +  e. 10, 20, +, 25, 15, -, *, 30, /</p> <p>4. Trace the following code, showing the contents of the 'STACK' after each invocation:</p>

	<ol style="list-style-type: none"> <li>1. PUSH(STACK, 1)</li> <li>2. PUSH(STACK, 2)</li> <li>3. PUSH(STACK, 3)</li> <li>4. POP(STACK, ITEM)</li> <li>5. PEEP(STACK, ITEM)</li> <li>6. PUSH(STACK, 4)</li> <li>7. POP(STACK, ITEM)</li> <li>8. POP(STACK, ITEM)</li> <li>9. PEEP(STACK, ITEM)</li> <li>10. PUSH(STACK, 5)</li> </ol>
	<p>5. Trace the following code, showing the contents of the Queue 'q' after each call:</p> <ol style="list-style-type: none"> <li>1. ENQUEUE(QUEUE, A)</li> <li>2. ENQUEUE(QUEUE, B)</li> <li>3. ENQUEUE(QUEUE, C)</li> <li>4. DEQUEUE(QUEUE, ITEM)</li> <li>5. DEQUEUE(QUEUE, ITEM)</li> <li>6. ENQUEUE(QUEUE, D)</li> <li>7. DEQUEUE(QUEUE, ITEM)</li> <li>8. DEQUEUE(QUEUE, ITEM)</li> <li>9. ENQUEUE(QUEUE, E)</li> <li>10. DEQUEUE(QUEUE, ITEM)</li> </ol>
	<p>6. Consider the following circular queue capable of accommodating a maximum of seven elements:</p> <p>Queue is: -, A, B, C, D, -, -, and Front = 2, Rear = 5</p> <p>Demonstration of the circular queue as the following operation take place:</p> <ol style="list-style-type: none"> <li>1. Enqueue "E"</li> <li>2. Enqueue "F"</li> <li>3. Dequeue</li> <li>4. Enqueue "G"</li> <li>5. Dequeue</li> <li>6. Enqueue "H"</li> </ol>
	<p>7. Suppose a queue is maintained by a circular array QUEUE with N = 12 memory cells. Find the number of elements in QUEUE if...</p> <ol style="list-style-type: none"> <li>a) FRONT = 4 and REAR = 8;</li> <li>b) FRONT = 10 and REAR = 3; and FRONT = 5 and REAR = 6; then two elements are deleted.</li> </ol>