

## CSE329:PRELUDE TO COMPETITIVE CODING

L:2 T:0 P:1 Credits:3

**Course Outcomes:** Through this course students should be able to

CO1 :: relate the theoretical as well as practical knowledge to form an amalgamation of working code

CO2 :: employ a working combination of solution to ubiquitous problems which are time and space efficient

CO3 :: identify and comprehend the inner workings behind the design of an optimal solution

CO4 :: illustrate the usage of algorithms and data structures in the design of an optimal solution towards a problem

CO5 :: recall the knowledge obtained from various algorithmic paradigms to formulate optimal solutions to real world problems

CO6 :: examine and utilise knowledge to build and design reliable code which is capable of passing various test cases

### Unit I

**Basic Data Structures :** Arrays, Declaring and processing 1D and 2D arrays, Insertion in an array, Deletion from array, Array rotations, Array arrangement, rearrangement, Matrix multiplication, Lower triangular and upper triangular matrix of array, Different operations on Matrices, Print a matrix in spiral form, Find distinct elements common to all rows in a matrix, String declaration and manipulation, K maximum sum from two arrays, Missing characters to make a string Pangram, Rearrange characters so that no two adjacent characters are same, Remove minimum number of characters so that two strings become anagram, Creation of stack using arrays, Creation of arrays using linked list, Creation of queue using array, Creation of queue using linked list, Implement two Stacks in an array, Implement Stack using Queues, Design a stack with operations on middle element, Implementation of Deque using circular array, Circular Queue, Reversing a Queue, Two pointer technique depicted by problem to find if there exists a pair in an unsorted array whose sum is equal to X, Count subarrays having an equal sum of elements at even and odd positions, Finding a peak element in array, Find subarray of length K with maximum Peak, Replace every element of the array with the previous element, Rearrange positive and negative numbers, Rearrange array such that even index elements are smaller and odd index elements are greater, Find the first non-repeating element in a given array of integers, Find the majority element, Count strings with consecutive ones, Check if all bits can be made same by single flip

### Unit II

**Basic math operations (addition, subtraction, multiplication, division and exponentiation) :** Fast modulo multiplication, Exponential squaring, N-th non-square number, Modular Exponentiation, Modular multiplicative inverse, Sum of middle row and element in matrix, Checking if all rows of a matrix are circular rotations of each other

### Unit III

**GCD and Primality testing :** Basic Euclidean algorithm, Extended Euclidean algorithm, Total number of divisors of a number, Finding all prime factors of a number, K-jagged numbers, Stormer numbers, Finding the prime factors by taking the square root, P-smooth numbers in given ranges, Problems based on GCD and primality testing

### Unit IV

**Searching techniques :** Iterative and recursive binary search, Find the missing number, Search an element in a sorted and rotated array, Find the K most frequent words from a string, Find a pair with a given difference, Find a peak element, Problems based on searching techniques, Length of longest subarray having sum in given range l r, Print all subarrays with sum in a given range, Minimum time required to produce m items

### Unit V

**Greedy techniques :** Greedy problem solving paradigm, Locally optimal choice, Global optimal choice, Job Sequencing problem, Job Selection problem, Minimum product subset of an array, Maximum product subset of an array, Minimum sum of product of two arrays, Bin packing problem, Activity Selection problem, Fractional Knapsack problem, Connect n ropes with minimum cost, Coin change problem, Problems based on greedy techniques, Majority Element

### Unit VI

**Naive Pattern Search :** The ubiquitous naïve pattern search problem, Checking if two strings are rotations of each other, Check if a string is substring of another string, Problems based on naive pattern search, Splitting a string into substrings with suitable practice problems

### **List of Practicals / Experiments:**

#### **Practicals**

- Program to insert an element in array
- Program to delete an element from an array
- Program to perform array arrangement
- Program to print a matrix in spiral form
- Program to find K maximum sum from two arrays
- Program to find missing characters to make a string Pangram
- Program to create a stack using arrays
- Program to print the reverse of a queue
- Program to print peak element in array
- Program to print positive and negative numbers
- Program to print total divisors of a number
- Program to print K jagged numbers
- Program for finding minimum product subset of an array
- Program to perform activity selection problem
- Program to find a pair with a given difference

#### **References:**

1. DATA STRUCTURES AND ALGORITHMS : CONCEPTS, TECHNIQUES AND APPLICATIONS by G. A. V. PAI, MCGRAW HILL EDUCATION
2. CRACKING THE CODING INTERVIEW by GAYLE LAAKMANN MCDOWELL, CAREERCUP