### ****Java Programming Questions****

#### For FY

1. Print the string "Hello, World!".
2. Check if a number is even or odd.
3. Print numbers from 1 to 100 using a loop.
4. Print the sum of two integers.
5. Reverse a given integer.
6. Find the largest of three numbers.
7. Check if a number is divisible by both 5 and 7.
8. Find the factorial of a number using a loop.
9. Check if a string is a palindrome.
10. Count the number of vowels in a string.
11. Find the area of a rectangle.
12. Swap two numbers without using a third variable.
13. Print a right-angled triangle pattern using stars (\*).
14. Find the GCD of two numbers.
15. Determine whether a number is a perfect square.

#### For SY and TY

1. Reverse a string without using built-in methods.
2. Count the occurrence of a specific character in a string.
3. Check if a number is prime.
4. Generate the Fibonacci series up to a specified number.
5. Find the sum of all digits of a number.
6. Create a method that accepts a string and returns it in uppercase.
7. Convert a string to an integer.
8. Sort an array of integers in ascending order.
9. Write a method that checks if a string contains only digits.
10. Find the second largest number in an array.
11. Implement a basic calculator for addition, subtraction, multiplication, and division.
12. Check if a string contains only uppercase characters.
13. Write a program that checks if two strings are anagrams.
14. Implement a program that returns the largest prime factor of a given number.
15. Write a program to remove duplicate elements from an array.

**For MSc**

* 1. Matrix Multiplication
* Write a program to multiply two matrices.
* Ensure the matrices are compatible for multiplication and handle incompatible matrices appropriately.
  1. Find the first & second smallest number in an array.
  2. Write a menu driven program to perform the following operations on a square matrix. Write separate functions for each option.
     1. Check if the matrix is symmetric.
     2. Display the trace of the matrix (sum of diagonal elements).
     3. Check if the matrix is an upper triangular matrix.
     4. Check if the matrix is a lower triangular matrix
     5. Check if it is an identity matrix.
  3. Given an n x n matrix, write a Java program to sort the matrix row-wise and then column-wise. The program should first sort all rows individually and then sort all columns individually.
  4. You are given numBottles full water bottles. You can exchange numExchange empty water bottles from the market for one full water bottle. The operation of drinking a full water bottle turns it into an empty bottle.

Write a Java program to determine the maximum number of water bottles you can drink given the initial number of full bottles and the exchange rate.

**Input:**

**numBottles:** The initial number of full water bottles.

**numExchange**: The number of empty water bottles required to exchange for one full water bottle.

**Output:**

The maximum number of **water bottles you can drink**.

Examples:

* + 1. Input: numBottles = 9, numExchange = 3

Output: 13

Explanation: You can exchange 3 empty bottles to get 1 full water bottle.

Number of water bottles you can drink: 9 + 3 + 1 = 13.

* + 1. Input: numBottles = 15, numExchange = 4

Output: 19

Explanation: You can exchange 4 empty bottles to get 1 full water bottle.

Number of water bottles you can drink: 15 + 3 + 1 = 19.

* 1. Generate N-th Tribonacci Sequence  
     The Tribonacci sequence T(n) is defined as follows:

T(0) = 0

T(1) = 1

T(2) = 1

T(n+3) = T(n) + T(n+1) + T(n+2) for n >= 0

Given the initial part of the sequence, write a Java program to calculate and print the Tribonacci sequence up to the N-th Tribonacci number based on user input.

Examples:

* + 1. Input: n = 4

Output: 0, 1, 1, 2, 4

Explanation:

T(0) = 0

T(1) = 1

T(2) = 1

T(3) = 0 + 1 + 1 = 2

T(4) = 1 + 1 + 2 = 4

* + 1. Input: n = 7

Output: 0, 1, 1, 2, 4, 7, 13, 24

* 1. Find the first & second largest number in an array.
  2. Kth smallest element in a row-wise and column-wise sorted 2D array

Last Updated : 19 Jul, 2024

Given an n x n matrix, where every row and column is sorted in non-decreasing order. Find the kth smallest element in the given 2D array

Example:

Input:k = 3 and array =

10, 20, 30, 40

15, 25, 35, 45

24, 29, 37, 48

32, 33, 39, 50

Output: 20

Explanation: The 3rd smallest element is 20

Input:k = 7 and array =

10, 20, 30, 40

15, 25, 35, 45

24, 29, 37, 48

32, 33, 39, 50

Output: 30

Explanation: The 7th smallest element is 30

### ****Python Programming Questions****

#### For FY

1. Print "Hello, World!".
2. Check if a number is even or odd.
3. Find the largest of two numbers.
4. Reverse a string.
5. Calculate the area of a circle.
6. Find the sum of a list of numbers.
7. Count the number of occurrences of a specific element in a list.
8. Print the Fibonacci sequence up to n terms.
9. Check if a number is prime.
10. Print numbers from 1 to 100.
11. Find the factorial of a number.
12. Swap two variables.
13. Count the number of vowels in a string.
14. Print a pattern of stars in the shape of a right-angled triangle.
15. Find the square of a number.

#### For SY and TY

1. Reverse a string without using built-in methods.
2. Find the sum of digits in a number.
3. Check if a string is a palindrome.
4. Count the number of vowels and consonants in a string.
5. Generate the Fibonacci series using a loop.
6. Find the largest number in a list.
7. Write a program that converts a decimal number to binary.
8. Write a program to find the greatest common divisor (GCD) of two numbers.
9. Merge two sorted lists into a single sorted list.
10. Implement a basic calculator for addition, subtraction, multiplication, and division.
11. Create a function that checks if a string is a pangram.
12. Write a program to remove duplicates from a list.
13. Implement the bubble sort algorithm.
14. Count the number of words in a string.
15. Implement a program to find the factorial of a number using recursion.

For MSc

* 1. Write a Python script that:

Creates a package math\_operations with two modules: basic\_operations.py and advanced\_operations.py.

The basic\_operations.py module should contain functions for addition, subtraction, multiplication, and division.

The advanced\_operations.py module should contain functions for power and square root.

Import these modules in a main script and demonstrate their usage.

* 1. Write a Python program that:

Reads the content of a file named input.txt.

Writes the reversed content to another file named reversed.txt.

* 1. Create a class Vehicle with attributes brand and speed. Add a method show\_details() to display these details. Then, create a subclass Car that inherits from Vehicle and adds an attribute mileage. Use the super() method to call the show\_details() of the parent class inside the Car class's show\_details() method.
  2. Implement All types of Inheritances
  3. Write a Python program that reads a text file and counts how many times each word appears in the file. The program should display the word frequency in descending order.
  4. skills\_A = {'Python', 'Java', 'SQL'}

skills\_B = {'Python', 'HTML', 'CSS'}

* + 1. Write a Python program to find:
    2. Skills that are in both sets.
    3. Skills that are unique to skills\_A.
    4. The union of both skill sets.
  1. Write a Python function multiply(a, \*args) that multiplies a fixed argument a with a variable number of additional arguments. Return the product.

### ****C Programming Questions****

#### For FY

1. Print "Hello, World!".
2. Find the sum of two integers.
3. Check if a number is even or odd.
4. Find the largest of two numbers.
5. Swap two numbers without using a temporary variable.
6. Print the multiplication table of a given number.
7. Find the factorial of a number using a loop.
8. Reverse a given number.
9. Find the area of a rectangle.
10. Count the number of digits in a number.
11. Calculate the perimeter of a circle.
12. Sum the elements of an array.
13. Print a pattern of stars in a triangle shape.
14. Calculate the power of a number.
15. Check if a number is divisible by 3 and 5.

#### For SY and TY

1. Reverse a string without using built-in functions.
2. Check if a number is prime.
3. Generate the Fibonacci sequence up to n terms.
4. Implement the Euclidean algorithm to find the GCD of two numbers.
5. Find the sum of digits of a number.
6. Count the number of vowels in a string.
7. Sort an array using bubble sort.
8. Check if a string is a palindrome.
9. Convert a decimal number to binary.
10. Find the largest element in an array.
11. Remove duplicate elements from an array.
12. Write a program to check if a number is Armstrong.
13. Merge two arrays into one sorted array.
14. Implement a simple calculator.
15. Find the second largest number in an array.

**For MSc**

1. Write a C program to find the **largest** and **second largest** elements in an array. The program should take an array of integers as input and output both the largest and second largest elements.
2. Write a C program to sort an array using the **Insertion Sort** algorithm. The program should take an array of integers as input and output the sorted array in ascending order.
3. **Sorting rows of matrix** in ascending order followed by columns in descending order. (Arrays.sort(arr[i], Collections.reverseOrder());)

Given a matrix, sort the rows of matrix in ascending order followed by sorting the columns in descending order.

Examples :

` Input : a[3][3] = {{1, 2, 3},{4, 5, 6}, {7, 8, 9}};

Output : 7 8 9

4 5 6

1 2 3

1. Write a C program to sort an array using the **Counting Sort** algorithm. The program should take an array of integers as input and output the sorted array.
2. Write a C program to generate an **N x N Magic Square**, where N is an odd integer. A Magic Square is a square matrix in which the sum of every row, column, and both diagonals are the same. The program should take the value of N as input and output the generated magic square.
3. Find x **power** n with complexity log n.
4. Write a C program to perform **Topological Sorting on a Directed Acyclic Graph (DAG).** The program should take the number of vertices and edges as input, along with the directed edges, and output the Topological Sorting of the graph.
5. Write a C program to solve **the Tower of Hanoi problem**. The program should take the number of disks as input and display the steps required to move all the disks from the source rod to the destination rod, using the auxiliary rod, while following the rules of the game:
   1. Only one disk can be moved at a time.
   2. Each move consists of taking the top disk from one stack and placing it on top of another stack.
   3. No disk may be placed on top of a smaller disk.
6. Write a C program to sort an array using the **Heap Sort** algorithm.
7. Write a C program to perform **matrix multiplication**. The program should take two matrices as input, multiply them, and output the resulting matrix. Ensure that the dimensions of the matrices are compatible for multiplication.
8. Write a C program to solve the **Fractional Knapsack problem** using a greedy approach. The program should take the following as input:
   * 1. The number of items.
     2. The weights and values of the items.
     3. The capacity of the knapsack.

The program should output the maximum value that can be obtained by putting items into the knapsack, considering that you can take fractions of the items.

1. Write a C program to implement the **Binary Search algorithm**. The program should take a sorted array and a target value as input, and return the index of the target value in the array using binary search.
2. Write a C program to check whether a given number is a **Perfect Number**.

### ****HTML Questions****

#### For FY

1. Create a basic HTML page with a title.
2. Create an ordered list with three items.
3. Create an unordered list with five items.
4. Create a table with 2 rows and 2 columns.
5. Add an image to a webpage.
6. Create a hyperlink to a webpage.
7. Create a form with a text input field and a submit button.
8. Create a paragraph and apply a simple CSS style to it.
9. Display a video from YouTube using the <iframe> tag.
10. Create a form with radio buttons.
11. Add a comment to an HTML file.
12. Create a button that submits a form.
13. Create a webpage with a background color.
14. Create a webpage with a heading and a paragraph.
15. Create an HTML document with a title and a link to another page.

#### For SY and TY

1. Create a responsive navigation bar.
2. Add an external CSS stylesheet to a webpage.
3. Create a form with input fields for name, email, and a submit button.
4. Create a table with 3 rows and 3 columns, each containing data.
5. Create a list of hyperlinks in a navigation menu.
6. Implement an image carousel using HTML and CSS.
7. Create a contact form with validation using HTML5 attributes.
8. Create a form with a dropdown menu for selecting a country.
9. Create a webpage with a footer that contains contact information.
10. Create an HTML page with multiple sections using <section>.
11. Create a webpage that includes a video and an audio player.
12. Add a favicon to a webpage.
13. Create a table with borders and alternating row colors.
14. Create a webpage with embedded Google Maps.
15. Create a form with checkboxes and a submit button.

For MSc CS

* 1. Demonstrate Internal, External, Inline CSS.
  2. Write an HTML program to display the user's current location (latitude and longitude) using the **Geolocation API**. The program should include a button to get the current location, and when clicked, it should show the latitude and longitude in a paragraph below the button. If the location is not available, it should display an error message.
  3. Write an HTML program that implements the Drag and Drop functionality. The program should include the following:
     1. A draggable element (such as a box or an image).
     2. A drop zone where the element can be dropped.
     3. The program should allow the user to drag the element and drop it into the drop zone.
     4. After the element is dropped, a message should appear indicating that the element was successfully dropped.
  4. Write an HTML program that embeds both an audio and video file on a webpage. The program should include the following:
     1. An audio player with controls to play, pause, and adjust volume for an audio file of your choice.
     2. A video player with controls to play, pause, and adjust volume for a video file of your choice.
     3. Both the audio and video files should be properly embedded using the <audio> and <video> tags, respectively.
  5. Write an HTML program to create a webpage that allows the user to switch between light and dark themes. The program should include the following features:
     1. A toggle button or switch to change between the two themes.
     2. The light theme should have a light background color (e.g., white) with dark text, and the dark theme should have a dark background color (e.g., black) with light text.
     3. Use CSS to define the styles for both light and dark themes and JavaScript to toggle between the themes when the button or switch is clicked.
  6. Write an HTML and CSS program to demonstrate the CSS Box Model. The program should include the following:
     1. Create a div element with the class box.
     2. Apply different properties to demonstrate the various parts of the box model:
     3. Content: Set a width and height for the box.
     4. Padding: Add padding inside the box to create space between the content and the border.
     5. Border: Add a border around the box.
     6. Margin: Add a margin outside the box to create space between the box and other elements.
  7. Write an HTML and CSS program to create a horizontal navigation bar with the following features:
     1. The navigation bar should contain links to "Home", "About", "Services", and "Contact".
     2. Style the navigation bar with a background color, and make the links change color when hovered over.
     3. Ensure the navigation bar is centered on the page and has some padding and margin.
  8. Write an HTML program to create a simple form that includes the following fields:
     1. A text input for the user's name.
     2. A radio button to select the gender.
     3. A checkbox to agree to terms and conditions.
     4. A submit button.
     5. Apply some CSS to style the form, including spacing between the fields, and make the submit button look visually appealing.