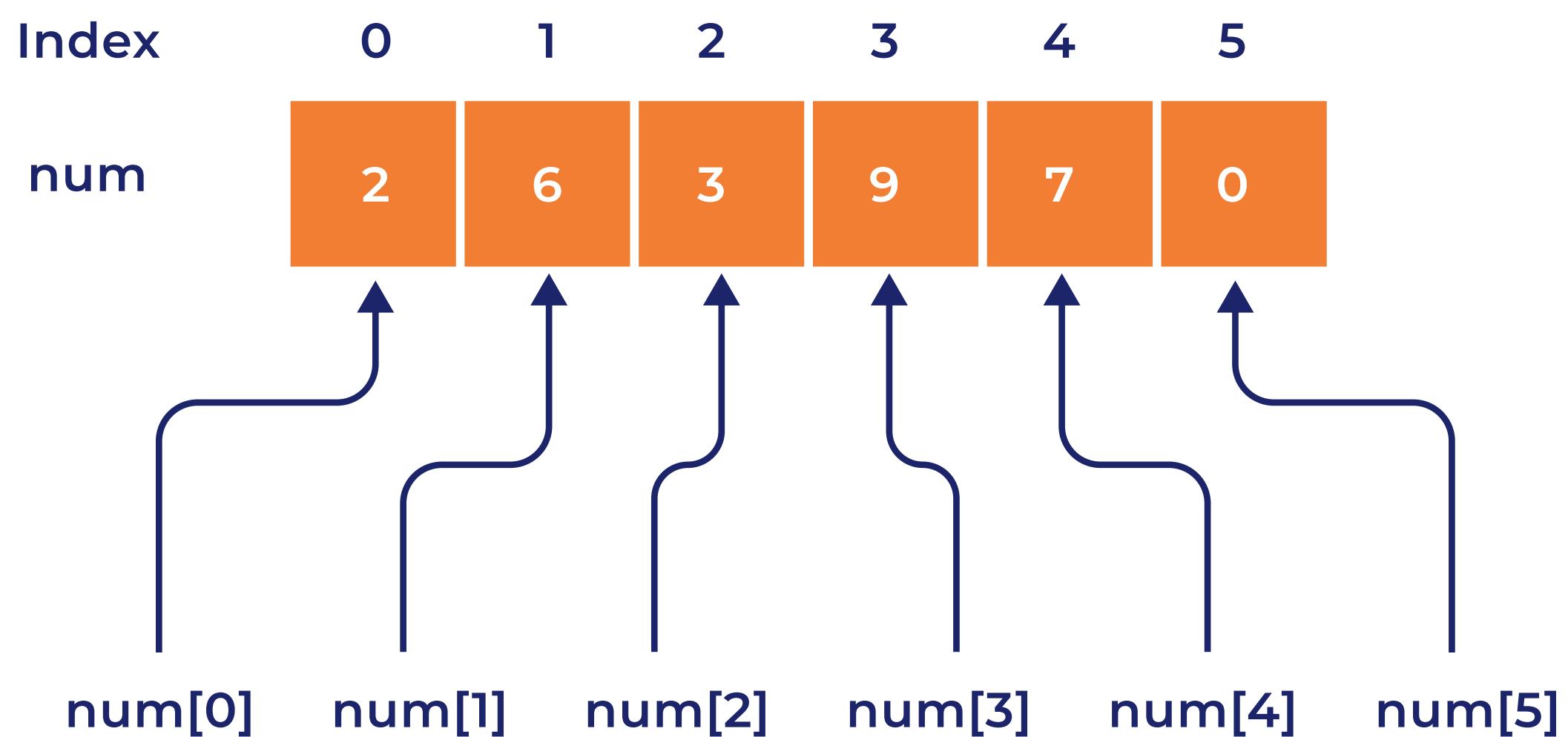


Master

ARRAYS

in 14 Days





Disclaimer

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.



DAY 1

Topic 1 - Introduction to Arrays

- Arrays are a fundamental data structure that allows storing a fixed-size sequence of elements of the same type.
- Arrays have a contiguous block of memory where elements are stored, and each element can be accessed using an index.
- Indexing starts from 0, so the first element is at index 0, the second at index 1, and so on.

EXAMPLE PROBLEM:

Write a program to create an array of all multiples of 5 till 50

```
#include <iostream>

int main() {
    const int arraySize = 10;      // Define the size of the array
    int multiples[arraySize];    // Declare an array to store the multiples

    int index = 0;
    for (int num = 5; num <= 50; num += 5) {
        multiples[index] = num;    // Assign each multiple to the array
        index++;
    }
    std::cout << "Array of multiples of 5 till 50: ";
    for (int i = 0; i < arraySize; i++) {
        std::cout << multiples[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```



DAY 2

Practice Questions:

1. Write a program to create an array of all prime number till 37 (think about the size that you would need to define?)

2. Create an array of strings to store the first names of 10 people. Write a program to find the 1st, 5th and 10th person's name



DAY 3

Topic 2 - Iterating through Arrays

For Loop Iteration

- The most common way to iterate through an array is by using a for loop.
- The loop variable is typically used as an index to access each element of the array.
- The loop continues until the index reaches the length of the array.

EXAMPLE PROBLEM:

Given an array of integers, write a program to calculate the sum of all elements.

```
int calculateSum(int arr[], int size) {  
    int sum = 0;  
  
    for (int i = 0; i < size; i++){  
        sum += arr[i];  
    }  
    return sum;  
}
```



DAY 4

Practice Questions:

1. Write a program to find the maximum and minimum elements in an array

2. Write a program to count the number of even numbers in an array of integers.



DAY 5

Topic 3 - Linear Search

- Linear search is a simple search algorithm used to find the position of a target value within an array or list.
- It sequentially checks each element of the array until a match is found or the entire array is traversed.
- The time complexity of linear search is $O(n)$, where n is the number of elements in the array.
- In the worst-case scenario, linear search may have to iterate through the entire array.

EXAMPLE PROBLEM:

Given an array of integers, write a program to search for a specific element and return its index if found, or -1 if not found.

```
int linearSearch(int arr[], int size, int target){  
    for (int i = 0; i < size; i++) {  
        if(arr[i] == target) {  
            return i; // Element found, return its index  
        }  
    }  
    return -1; // Element not found  
}
```



DAY 6

Practice Questions:

1. Implement a function to check if an array of characters contains a specific character using linear search.

2. Write a program to find the smallest positive element in an array of integers using linear search.



DAY 7

Topic 4 - Binary Search

- Binary search is an efficient search algorithm used to find a target value within a sorted array.
- It follows a divide-and-conquer approach by repeatedly dividing the array into halves and narrowing down the search range.
- It has better time complexity than linear search that is $O(\log n)$ where n is number of elements in the array

EXAMPLE PROBLEM:

Given a sorted array of integers, write a program to search for a specific element using binary search and return its index if found, or -1 if not found.

```
int binarySearch(int arr[], int size, int target) {  
    int left = 0;  
    int right = size - 1;  
  
    while (left <= right) {  
        int mid = left + (right - left) / 2;  
  
        if (arr[mid] == target) {  
            return mid; // Element found, return its index  
        } else if (arr[mid] < target) {  
            left = mid + 1; // Target is in the right half  
        } else {  
            right = mid - 1; // Target is in the left half  
        }  
    }  
  
    return -1; //Element not found  
}
```



DAY 8

Practice Questions:

1. Write a program to find the square root of a given number using binary search.
2. Write a program to find the first occurrence of a specific element in a sorted array.



DAY 9

Topic 5 - Intro to Multidimensional Arrays

- A multidimensional array is an array that contains one or more arrays as its elements.
- It represents data in multiple dimensions, such as rows and columns, or matrices.
- Elements in a multidimensional array can be accessed using multiple indices corresponding to each dimension.

EXAMPLE PROBLEM:

Write a program to create a 3*3 matrix

```
#include <iostream>

int main() {
    const int rows = 3;
    const int cols = 3;
    int matrix[rows][cols];

    // Getting input for matrix elements
    std::cout << "Enter the elements of the matrix:" << std::endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            std::cout << "Enter element at position (" << i << ", " << j << "): ";
            std::cin >> matrix[i][j];
        }
    }

    // Printing the matrix
    std::cout << "Matrix:" << std::endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            std::cout << matrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
    return 0;
}
```



DAY 10

Practice Questions:

1. Write a program to create a tic tac toe board to represent any winning state of “O” (Hint: use a character array)

2. Write a program to calculate the sum of all elements in a 2D array.

3. Write a program to add, subtract, multiply two matrices represented by 2D arrays.



DAY 11

Topic 6 - Sorting Algorithms

1. Bubble Sort

Works by swapping adjacent elements in repeated passes, if they are not in correct order. High time complexity and not suitable for large datasets.

Time Complexity: $O(N^2)$

1. Insertion Sort

The array is split into sorted and unsorted parts. Unsorted elements are picked and placed at their correct position in the sorted part.

Time Complexity: $O(N^2)$

1. Selection Sort

The smallest value among the unsorted elements of the array is selected in every pass and inserted to its appropriate position into the array.

Time Complexity: $O(N^2)$

Example Problem:

Sort an array of integers in ascending order using a sorting algorithm of your choice.

Using selection sort



EXAMPLE PROBLEM:

Sort an array of integers in ascending order using a sorting algorithm of your choice.

Using selection sort

```
void selectionSort(int arr[], int n)
{
    int i, j, minidx;

    // One by one move boundary of unsorted subarray
    for (i = 0; i < n - 1; i++) {

        // Find the minimum element in the unsorted array
        minidx = i;
        for (j = i + 1; j < n; j++) {
            if (arr[j] < arr[minidx])
                minidx = j;
        }

        // Swap the found minimum element with the first element
        if (minidx != i)
            swap(arr[minidx], arr[i]);
    }
}
```



DAY 12

Practice Questions:

1. Implement a function to sort an array of floating-point numbers in descending order using a sorting algorithm.
2. Write a program to find the kth smallest element in an unsorted array.



DAY 13

Arrays Based Practice Problems

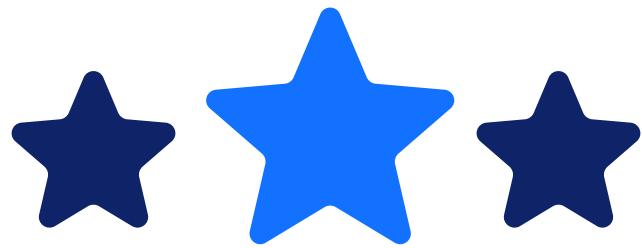
1. <https://practice.geeksforgeeks.org/problems/kadane-algorithm-1587115620/1>
2. <https://leetcode.com/problems/next-permutation/>
3. <https://leetcode.com/problems/two-sum/>
4. <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>
5. <https://leetcode.com/problems/product-of-array-except-self/>
6. <https://leetcode.com/problems/maximum-subarray/>
7. <https://leetcode.com/problems/contains-duplicate/>
8. <https://leetcode.com/problems/container-with-most-water/>
9. <https://leetcode.com/problems/3sum/>
10. <https://leetcode.com/problems/maximum-product-subarray/>



DAY 14

2D Arrays Based Practice Problems

1. <https://leetcode.com/problems/spiral-matrix/>
2. <https://leetcode.com/problems/set-matrix-zeroes/>
3. <https://leetcode.com/problems/rotate-image/>
4. <https://leetcode.com/problems/valid-sudoku/>
5. <https://practice.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1>
6. <https://www.geeksforgeeks.org/common-elements-in-all-rows-of-a-given-matrix/>
7. <https://www.geeksforgeeks.org/find-a-specific-pair-in-matrix/>



WHY BOSSCODER?

 **1000+** Alumni placed at Top Product-based companies.

 More than **136% hike** for every **2 out of 3** working professional.

 Average package of **24LPA**.

The syllabus is most up-to-date and the list of problems provided covers all important topics.

Lavanya
 Meta



Course is very well structured and streamlined to crack any MAANG company

Rahul
 Google



[EXPLORE MORE](#)