

Arrays

1.) Find common elements between two arrays

```
import java.util.HashSet;
import java.util.Set;

public class CommonElements {

    public static void main(String[] args) {
        int[] array1 = {1, 2, 3, 4, 5};
        int[] array2 = {4, 5, 6, 7, 8};
        Set<Integer> commonElements = findCommonElements(array1,
array2);

        System.out.println("Common elements: " + commonElements);
    }

    public static Set<Integer> findCommonElements(int[] array1,
int[] array2) {
        Set<Integer> set1 = new HashSet<>();
        Set<Integer> commonSet = new HashSet<>();
        // Add elements of the first array to the set
        for (int num : array1) {
            set1.add(num);
        }
        // Check for common elements in the second array
        for (int num : array2) {
            if (set1.contains(num)) {
                commonSet.add(num);
            }
        }
        return commonSet;
    }
}
```

Input: array1 = {1,2,3,4,5} and

array2 = {4,5,6,7,8}

Output: Common elements: [4, 5]

2.) Find first and last element of ArrayList

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("Apple");
        arrayList.add("Banana");
        arrayList.add("Cherry");
        arrayList.add("Date");
        arrayList.add("Elderberry");
        if (!arrayList.isEmpty()) {

            String firstElement = arrayList.get(0);
            String lastElement = arrayList.get(arrayList.size() - 1);

            System.out.println("First element: " + firstElement);
            System.out.println("Last element: " + lastElement);
        } else {
            System.out.println("The ArrayList is empty.");
        }
    }
}
```

Output:

First element: Apple

Last element: Elderberry

3.) Sort an array without using in-built method

```
public class Main {  
    public static void main(String[] args) {  
        int[] array = {5, 2, 9, 1, 6};  
        selectionSort(array);  
  
        System.out.println("Sorted array:");  
        for (int num : array) {  
            System.out.print(num + " ");  
        }  
    }  
  
    public static void selectionSort(int[] array) {  
        int n = array.length;  
        for (int i = 0; i < n - 1; i++) {  
            int minIndex = i;  
            for (int j = i + 1; j < n; j++) {  
                if (array[j] < array[minIndex]) {  
                    minIndex = j;  
                }  
            }  
  
            // Swap array[i] and array[minIndex]  
            int temp = array[i];  
            array[i] = array[minIndex];  
            array[minIndex] = temp;  
        }  
    }  
}
```

Output:

Sorted array:

1 2 5 6 9

4.) Remove duplicates from an Array

```
import java.util.HashSet;
import java.util.Set;

public class Main {

    public static void main(String[] args) {
        int[] array = {5, 2, 9, 1, 6, 2, 5};

        int[] uniqueArray = removeDuplicates(array);

        System.out.println("Array with duplicates removed:");
        for (int num : uniqueArray) {
            System.out.print(num + " ");
        }

        public static int[] removeDuplicates(int[] array) {
            Set<Integer> set = new HashSet<>();
            for (int num : array) {
                set.add(num);
            }

            int[] result = new int[set.size()];
            int i = 0;
            for (int num : set) {
                result[i++] = num;
            }

            return result;
        }
    }
}
```

Output:

Array with duplicates removed:

1 2 5 6 9

5.) Remove duplicates from an ArrayList

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arrayList = new ArrayList<>();
        arrayList.add(5);
        arrayList.add(2);
        arrayList.add(9);
        arrayList.add(1);
        arrayList.add(6);
        arrayList.add(2);
        arrayList.add(5);
        ArrayList<Integer> uniqueList =
removeDuplicates(arrayList);

        System.out.println("ArrayList with duplicates
removed:");
        for (int num : uniqueList) {
            System.out.print(num + " ");
        }
    }
    public static ArrayList<Integer>
removeDuplicates(ArrayList<Integer> list) {
        Set<Integer> set = new HashSet<>(list);
        return new ArrayList<>(set);
    }
}
```

Output:

ArrayList with duplicates removed:

1 2 5 6 9

6.) Find the missing number in an Array

```
public class Main {  
    public static void main(String[] args) {  
        int[] array = {1, 2, 4, 5, 6}; // Missing number is 3  
        int missingNumber = findMissingNumber(array);  
        System.out.println("The missing number is: " + missingNumber);  
    }  
  
    public static int findMissingNumber(int[] array) {  
        int n = array.length + 1; // Since one number is missing, the length  
        // should be n+1  
        int totalSum = n * (n + 1) / 2; // Sum of first n natural numbers  
        int arraySum = 0;  
        for (int num : array) {  
            arraySum += num;  
        }  
        return totalSum - arraySum;  
    }  
}
```

Output:

The missing number is: 3

7.) Find the largest and smallest element in an Array

```
public class Main {  
    public static void main(String[] args) {  
        int[] array = {5, 2, 9, 1, 6, 3};  
  
        int[] result = findLargestAndSmallest(array);  
  
        System.out.println("Smallest element: " + result[0]);  
        System.out.println("Largest element: " + result[1]);  
    }  
  
    public static int[] findLargestAndSmallest(int[] array) {  
        if (array == null || array.length == 0) {  
            throw new IllegalArgumentException("Array must not be null or  
empty");  
        }  
  
        int smallest = array[0];  
        int largest = array[0];  
  
        for (int num : array) {  
            if (num < smallest) {  
                smallest = num;  
            }  
            if (num > largest) {  
                largest = num;  
            }  
        }  
        return new int[]{smallest, largest};  
    }  
}
```

Output:

Smallest element: 1

Largest element: 9

8.) Search element in an Array

```
public class Main {  
    public static void main(String[] args) {  
        int[] array = {5, 2, 9, 1, 6, 3};  
        int target = 6;  
  
        int index = linearSearch(array, target);  
  
        if (index != -1) {  
            System.out.println("Element " + target + " found at index: " +  
index);  
        } else {  
            System.out.println("Element " + target + " not found in the  
array.");  
        }  
    }  
    public static int linearSearch(int[] array, int target) {  
        for (int i = 0; i < array.length; i++) {  
            if (array[i] == target) {  
                return i; // Element found, return index  
            }  
        }  
        return -1; // Element not found  
    }  
}
```

Output:

Element 6 found at index: 4

Element 10 not found in the array

9.) Array consists of integers and special characters, sum only integers

```
public class Main {  
    public static void main(String[] args) {  
        String[] array = {"5", "2", "9", "a", "1", "6", "#", "3"};  
  
        int sum = sumIntegers(array);  
  
        System.out.println("Sum of integers in the array: " + sum);  
    }  
  
    public static int sumIntegers(String[] array) {  
        int sum = 0;  
        for (String element : array) {  
            try {  
                int num = Integer.parseInt(element);  
                sum += num;  
            } catch (NumberFormatException e) {  
                // Ignore non-integer elements  
            }  
        }  
        return sum;  
    }  
}
```

Output:

Sum of integers in the array: 26

10.) Find Minimum and Maximum from an Array

```
public class Main {
    public static void main(String[] args) {
        int[] array = {5, 2, 9, 1, 6, 3};

        // Find maximum and minimum
        int max = findMaximum(array);
        int min = findMinimum(array);

        // Print the results
        System.out.println("Minimum value in the array: " + min);
        System.out.println("Maximum value in the array: " + max);
    }

    public static int findMaximum(int[] array) {
        if (array.length == 0) {
            throw new IllegalArgumentException("Array must not be empty");
        }
        int max = array[0]; // Initialize max to the first element
        for (int i = 1; i < array.length; i++) {
            if (array[i] > max) {
                max = array[i]; // Update max if current element is larger
            }
        }
        return max;
    }

    public static int findMinimum(int[] array) {
        if (array.length == 0) {
            throw new IllegalArgumentException("Array must not be empty");
        }
        int min = array[0]; // Initialize min to the first element
        for (int i = 1; i < array.length; i++) {
            if (array[i] < min) {
                min = array[i]; // Update min if current element is smaller
            }
        }
        return min;
    }
}
```

Output:

Minimum value in the array: 1

Maximum value in the array: 9

11.) Java program to count Odd and Even number from given array

Input: {1,2,3,4,5,6,7,8,9}

```
public class Main {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        int[] count = countOddAndEven(array);

        System.out.println("Even numbers count: " + count[1]);
        System.out.println("Odd numbers count: " + count[0]);
    }

    public static int[] countOddAndEven(int[] array) {
        int[] count = new int[2]; // Index 0 for odd count, Index 1 for
even count

        for (int num : array) {
            if (num % 2 == 0) {
                count[1]++; // Increment even count
            } else {
                count[0]++; // Increment odd count
            }
        }
        return count;
    }
}
```

Output:

Even numbers count: 4
Odd numbers count: 5

12.) Java program – input array was
given [1,1,2,2,3,4,5,5,6,6],
Output – [3,4]

```
import java.util.HashMap;
import java.util.Map;
import java.util.ArrayList;
import java.util.List;
public class Main {

    public static void main(String[] args) {
        int[] array = {1, 1, 2, 2, 3, 4, 5, 5, 6, 6};
        List<Integer> result = findNonRepeatedElements(array);
        System.out.println("Non-repeated elements: " + result);
    }

    public static List<Integer> findNonRepeatedElements(int[]
array) {

        // Step 1: Count occurrences of each element using a
        // HashMap
        Map<Integer, Integer> countMap = new HashMap<>();
        for (int num : array) {
            countMap.put(num, countMap.getOrDefault(num, 0) + 1);
        }
        // Step 2: Identify elements with count equal to 1 (non-
        // repeated)
        List<Integer> nonRepeatedElements = new ArrayList<>();
        for (Map.Entry<Integer, Integer> entry :
countMap.entrySet()) {
            if (entry.getValue() == 1) {
                nonRepeatedElements.add(entry.getKey());
            }
        }
        return nonRepeatedElements;
    }
}
```

Output :
Non-repeated elements: [3, 4]

Java program to implement **hashCode** and **equals**

```
import java.util.Objects;

public class Student {
    private int id;
    private String name;

    // Constructor
    public Student(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // Getters and setters (omitted for brevity)

    // hashCode method
    @Override
    public int hashCode() {
        return Objects.hash(id, name);
    }

    // equals method
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null || getClass() != obj.getClass())
            return false;
        Student student = (Student) obj;
        return id == student.id && Objects.equals(name, student.name);
    }

    public static void main(String[] args) {
        // Creating objects of Student class
        Student student1 = new Student(1, "Alice");
        Student student2 = new Student(2, "Bob");
        Student student3 = new Student(1, "Alice");

        // Testing equals method
        System.out.println("student1.equals(student2): " +
            student1.equals(student2)); // Output: false
        System.out.println("student1.equals(student3): " +
            student1.equals(student3)); // Output: true

        // Testing hashCode method
        System.out.println("HashCode of student1: " + student1.hashCode());
        System.out.println("HashCode of student2: " + student2.hashCode());
        System.out.println("HashCode of student3: " + student3.hashCode());
    }
}
```