# *ASSIGNMENT-1 (ARRAYS)*

*SOLUTIONS:-*

*Q1.*

```java
import java.util.Scanner;


public class arrays_assig1 {

    static void findIndices(int[] arr, int target) {
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                if (arr[i] + arr[j] == target) {
                    System.out.println(i + " " + j);
                }
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.print("Enter the elements of array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the target you want to find: ");
        int target = sc.nextInt();
        findIndices(arr, target);
    }
}
```

*Q2.*

```java
public class arrays_assig2 {
    public static int removeElement(int[] nums, int val) {
        int k = 0;
        for (int i = 0; i < nums.length; i++) {
            if (nums[i] != val) {
                nums[k] = nums[i];
                k++;
            }
        }
```

```
        return k;
    }

    public static void main(String[] args) {
        int[] nums = { 3, 2, 2, 3 };
        int val = 3;
        int k = removeElement(nums, val);
        System.out.println("k = " + k);
        System.out.print("nums = [");
        for (int i = 0; i < k; i++) {
            System.out.print(nums[i]);
            if (i < k - 1) {
                System.out.print(", ");
            }
        }
        System.out.println(", _*, _*]");
    }
}
```

*Q3.*

```
import java.util.*;

public class arrays_assig3 {

    static int findIndices(int[] arr, int target) {
        int n = arr.length;
        int low = 0;
        int high = n - 1;
        int index = 0;
        while (low <= high) {
            int mid = low + (high - low) / 2;
            if (arr[mid] == target) {
                return mid;
            } else if (arr[mid] > target) {
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }
        return low;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of array: ");
        int n = sc.nextInt();
```

```
        int[] arr = new int[n];
        System.out.print("Enter the elements of array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the target you want to find: ");
        int target = sc.nextInt();
        int index = findIndices(arr, target);
        System.out.println("Index where element is found : " + index);

    }
}
```

*Q4.*

```
public class arrays_assig4 {
    public static int[] plusOne(int[] digits) {
        int n = digits.length;

        for (int i = n - 1; i >= 0; i--) {
            if (digits[i] < 9) {
                digits[i]++;
                return digits;
            } else {
                digits[i] = 0;
            }
        }

        int[] newDigits = new int[n + 1];
        newDigits[0] = 1;
        return newDigits;
    }

    public static void main(String[] args) {
        int[] digits = { 1, 2, 9 };
        int[] result = plusOne(digits);
        for (int i = 0; i < result.length; i++) {
            System.out.print(result[i] + " ");
        }
    }
}
```

*Q5.*

```
public class arrays_assig5 {
    public static void merge(int[] nums1, int m, int[] nums2, int n) {
        int i = m - 1;
```

```
        int j = n - 1;
        int k = m + n - 1;

        while (i >= 0 && j >= 0) {
            if (nums1[i] >= nums2[j]) {
                nums1[k] = nums1[i];
                i--;
            } else {
                nums1[k] = nums2[j];
                j--;
            }
            k--;
        }

        while (j >= 0) {
            nums1[k] = nums2[j];
            j--;
            k--;
        }
    }

    public static void main(String[] args) {
        int[] nums1 = { 1, 2, 3, 0, 0, 0 };
        int m = 3;
        int[] nums2 = { 2, 5, 6 };
        int n = 3;
        merge(nums1, m, nums2, n);
        for (int i = 0; i < m + n; i++) {
            System.out.print(nums1[i] + " ");

        }

    }
}
```

*Q6.*

```
public class arrays_assig6 {

    static boolean checkRepeating(int[] nums) {
        for (int i = 0; i < nums.length; i++) {
            for (int j = i + 1; j < nums.length; j++) {
                if (nums[i] == nums[j]) {
                    return true;
                }
            }
        }
```

```
        return false;
    }

    public static void main(String[] args) {
        int[] nums = { 1, 2, 3, 1 };
        boolean result = checkRepeating(nums);
        System.out.println(result);
    }
}
```

*Q7.*

```
public class arrays_assig7 {
    public static void moveZeroes(int[] nums) {
        int n = nums.length;
        int insertPos = 0;

        // Move non-zero elements to the front of the array
        for (int num : nums) {
            if (num != 0) {
                nums[insertPos] = num;
                insertPos++;
            }
        }

        // Fill the remaining positions with zeros
        while (insertPos < n) {
            nums[insertPos] = 0;
            insertPos++;
        }
    }

    public static void main(String[] args) {
        int[] nums = { 0, 1, 0, 3, 12 };
        moveZeroes(nums);
        for (int i = 0; i < nums.length; i++) {
            System.out.print(nums[i] + " ");

        }
    }
}
```

*Q8.*

```
import java.util.Arrays;
```

```java
public class arrays_assig8 {

    static int[] findErrorNums(int[] nums) {
        int N = nums.length, sum = N * (N + 1) / 2;
        int[] ans = new int[2];
        boolean[] seen = new boolean[N + 1];
        for (int num : nums) {
            sum -= num;
            if (seen[num])
                ans[0] = num;
            seen[num] = true;
        }
        ans[1] = sum + ans[0];
        return ans;
    }

    public static void main(String[] args) {
        int[] nums = { 1, 2, 2, 4 };
        int[] result = findErrorNums(nums);
        System.out.println(Arrays.toString(result));
    }
}
```