1.https://leetcode.com/problems/merge-sorted-array/

```
public class MergeSortedArray {

  public static void main(String[] args) {
    int[] nums1 = {1, 2, 3, 0, 0, 0};
    int[] nums2 = {2, 5, 6};
    int m = 3;
    int n = 3;

    int[] mergedArray = mergeSortedArray(nums1, nums2, m, n);

    for (int i = 0; i < mergedArray.length; i++) {
      System.out.print(mergedArray[i] + " ");
    }
  }

  public static int[] mergeSortedArray(int[] nums1, int[] nums2, int m, int n) {
    int[] mergedArray = new int[m + n];
```

```
int i = 0;
int j = 0;
int k = 0;

while (i < m && j < n) {
  if (nums1[i] <= nums2[j]) {
    mergedArray[k] = nums1[i];
    i++;
  } else {
    mergedArray[k] = nums2[j];
    j++;
  }
  k++;
}

while (i < m) {
  mergedArray[k] = nums1[i];
  i++;
  k++;
}

while (j < n) {
  mergedArray[k] = nums2[j];
  j++;
  k++;
```

```
    }

    return mergedArray;
  }
}
```

2.https://practice.geeksforgeeks.org/problems/consecutive-array-elements2711/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

```
public class AreConsecutives {

  public static void main(String[] args) {
    int[] arr = {1, 2, 3, 4, 5};
    int n = arr.length;

    boolean isConsecutive = areConsecutives(arr, n);
```

```
    System.out.println(isConsecutive);
  }

  public static boolean areConsecutives(int[] arr, int
n) {
    int minElement = arr[0];
    int maxElement = arr[0];

    for (int i = 1; i < n; i++) {
      minElement = Math.min(minElement, arr[i]);
      maxElement = Math.max(maxElement, arr[i]);
    }

    if (maxElement - minElement + 1 != n) {
      return false;
    }

    for (int i = 1; i < n; i++) {
      if (arr[i] != minElement + i - 1) {
        return false;
      }
    }

    return true;
  }
```

}

3.https://practice.geeksforgeeks.org/problems/chocolate-distribution-problem3825/1

```
import java.util.Arrays;

public class ChocolateDistribution {

    public static int chocolateDistribution(int n, int c) {
        // Sort the chocolates in increasing order.
        int[] chocolates = new int[n];
        for (int i = 0; i < n; i++) {
            chocolates[i] = i * c;
        }
        Arrays.sort(chocolates);

        // Find the minimum difference between the largest and smallest chocolates.
        int minWrappers = Integer.MAX_VALUE;
        for (int i = 0; i < n - 1; i++) {
            minWrappers = Math.min(minWrappers,
```

```
chocolates[i + 1] - chocolates[i]);
        }

        return minWrappers;
    }

    public static void main(String[] args) {
        int n = 5;
        int c = 2;
        System.out.println(chocolateDistribution(n, c));
    }
}
```

4.https://leetcode.com/problems/sort-an-array/description/
Bubble sort

```
import java.util.Arrays;

public class BubbleSort {

    public static void bubbleSort(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            for (int j = 0; j < array.length - i - 1; j++) {
```

```
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] array = {5, 2, 3, 1};
        bubbleSort(array);
        System.out.println(Arrays.toString(array));
    }
}
```

5.https://leetcode.com/problems/sort-colors/description/

```
import java.util.Arrays;

public class SortColors {
```

```
public static void sortColors(int[] nums) {
    int i = 0;
    int j = 0;
    int k = nums.length - 1;

    while (j <= k) {
        if (nums[j] == 0) {
            int temp = nums[i];
            nums[i] = nums[j];
            nums[j] = temp;
            i++;
            j++;
        } else if (nums[j] == 1) {
            j++;
        } else {
            int temp = nums[k];
            nums[k] = nums[j];
            nums[j] = temp;
            k--;
        }
    }
}

public static void main(String[] args) {
```

```
        int[] nums = {2, 0, 1, 1, 0, 2, 0, 1};
        sortColors(nums);
        System.out.println(Arrays.toString(nums));
    }
}
```

6.https://leetcode.com/problems/sort-an-array/description/

Selection sort

```
import java.util.Arrays;

public class SelectionSort {

    public static void selectionSort(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < array.length; j++) {
                if (array[j] < array[minIndex]) {
                    minIndex = j;
                }
```

```
            }

            int temp = array[i];
            array[i] = array[minIndex];
            array[minIndex] = temp;
        }
    }

    public static void main(String[] args) {
        int[] array = {5, 2, 3, 1};
        selectionSort(array);
        System.out.println(Arrays.toString(array));
    }
}
```