Answer 1)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class HelloWorld {
    public static int[][] convertArray(int[] original, int m, int n) {
        if (original.length != m * n) {
            return new int[0][0];
        }

        int[][] result = new int[m][n];
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                result[i][j] = original[i * n + j];
            }
        }

        return result;
    }

    public static void main(String[] args) {
        int[] original = {1, 2, 3, 4};
        int m = 2;
        int n = 2;

        int[][] output = convertArray(original, m, n);

        // Print the output 2D array
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(output[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Answer 2)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class HelloWorld {
```

```
    public static int arrangeCoins(int n) {
        int row = 1;
        while (n >= row) {
            n -= row;
            row++;
        }
        return row - 1;
    }

    public static void main(String[] args) {
        int n = 5;
        int completeRows = arrangeCoins(n);
        System.out.println("Number of complete rows: " + completeRows);
    }
}
```

Answer 3)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

import java.util.Arrays;

public class HelloWorld {
    public static int[] sortedSquares(int[] nums) {
        int n = nums.length;
        int[] result = new int[n];

        int left = 0;
        int right = n - 1;
        int index = n - 1;

        while (left <= right) {
            int leftSquare = nums[left] * nums[left];
            int rightSquare = nums[right] * nums[right];

            if (leftSquare > rightSquare) {
                result[index] = leftSquare;
                left++;
            } else {
                result[index] = rightSquare;
                right--;
```

```
            }

            index--;
        }

        return result;
    }

    public static void main(String[] args) {
        int[] nums = {-4, -1, 0, 3, 10};
        int[] result = sortedSquares(nums);

        System.out.println("Sorted Squares: " + Arrays.toString(result));
    }
}
```

Answer 4)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

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

public class HelloWorld {
    public static List<List<Integer>> findDisappearedNumbers(int[] nums1, int[] nums2) {
        Set<Integer> set1 = new HashSet<>();
        Set<Integer> set2 = new HashSet<>();

        for (int num : nums1) {
            set1.add(num);
        }

        for (int num : nums2) {
            set2.add(num);
        }

        List<Integer> distinctNums1 = new ArrayList<>();
        List<Integer> distinctNums2 = new ArrayList<>();
```

```
        for (int num : set1) {
            if (!set2.contains(num)) {
                distinctNums1.add(num);
            }
        }

        for (int num : set2) {
            if (!set1.contains(num)) {
                distinctNums2.add(num);
            }
        }

        List<List<Integer>> answer = new ArrayList<>();
        answer.add(distinctNums1);
        answer.add(distinctNums2);

        return answer;
    }

    public static void main(String[] args) {
        int[] nums1 = {1, 2, 3};
        int[] nums2 = {2, 4, 6};

        List<List<Integer>> answer = findDisappearedNumbers(nums1, nums2);

        System.out.println("Distinct integers in nums1 not in nums2: " + answer.get(0));
        System.out.println("Distinct integers in nums2 not in nums1: " + answer.get(1));
    }
}
```

Answer 5)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class HelloWorld {
    public static int distanceValue(int[] arr1, int[] arr2, int d) {
        int distance = 0;

        for (int num1 : arr1) {
            boolean found = false;
            for (int num2 : arr2) {
```

```
                if (Math.abs(num1 - num2) <= d) {
                    found = true;
                    break;
                }
            }
            if (!found) {
                distance++;
            }
        }

        return distance;
    }

    public static void main(String[] args) {
        int[] arr1 = {4, 5, 8};
        int[] arr2 = {10, 9, 1, 8};
        int d = 2;

        int distance = distanceValue(arr1, arr2, d);
        System.out.println("Distance value: " + distance);
    }
}
```

Answer 6)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

import java.util.ArrayList;
import java.util.List;

public class HelloWorld {
    public static List<Integer> findDuplicates(int[] nums) {
        List<Integer> duplicates = new ArrayList<>();

        for (int i = 0; i < nums.length; i++) {
            int index = Math.abs(nums[i]) - 1;

            if (nums[index] < 0) {
                duplicates.add(index + 1);
            } else {
                nums[index] = -nums[index];
```

```
            }
        }

        for (int i = 0; i < nums.length; i++) {
            nums[i] = Math.abs(nums[i]);
        }

        return duplicates;
    }

    public static void main(String[] args) {
        int[] nums = {4, 3, 2, 7, 8, 2, 3, 1};
        List<Integer> duplicates = findDuplicates(nums);

        System.out.println("Duplicates: " + duplicates);
    }
}
```

Answer 7)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

public class Main {
    public static int findMin(int[] nums) {
        int left = 0;
        int right = nums.length - 1;

        while (left < right) {
            int mid = left + (right - left) / 2;

            if (nums[mid] > nums[right]) {
                left = mid + 1;
            } else {
                right = mid;
            }
        }

        return nums[left];
    }

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

```
        int[] nums = {3, 4, 5, 1, 2};
        int min = findMin(nums);

        System.out.println("Minimum element: " + min);
    }
}
```

Answer 8)

```
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Main {
    public static int[] findOriginalArray(int[] changed) {
        if (changed.length % 2 != 0) {
            return new int[0]; // If the length is odd, it can't be a doubled array
        }

        Map<Integer, Integer> countMap = new HashMap<>();

        for (int num : changed) {
            countMap.put(num, countMap.getOrDefault(num, 0) + 1);
        }

        List<Integer> originalList = new ArrayList<>();

        for (int num : changed) {
            if (countMap.getOrDefault(num, 0) == 0) {
                continue; // Skip if the number has already been used
            }

            int half = num / 2;
            int count = countMap.getOrDefault(half, 0);

            if (num % 2 == 0 && count > 0) {
                originalList.add(half);
                countMap.put(half, count - 1);
```

```
            } else {
                return new int[0]; // If the number is odd or the half is not found, it's not a valid doubled
array
            }

            countMap.put(num, countMap.get(num) - 1);
        }

        int[] originalArray = new int[originalList.size()];

        for (int i = 0; i < originalList.size(); i++) {
            originalArray[i] = originalList.get(i);
        }

        return originalArray;
    }

    public static void main(String[] args) {
        int[] changed = {1, 3, 4, 2, 6, 8};
        int[] original = findOriginalArray(changed);

        System.out.print("Original array: ");
        if (original.length == 0) {
            System.out.println("[]");
        } else {
            for (int num : original) {
                System.out.print(num + " ");
            }
            System.out.println();
        }
    }
}
```