Problem 1

public class FindTwoElements {

static int[] findTwoElement(int arr[], int n) {

int repeating = -1, missing = -1;

for (int i = 0; i < n; i++) {

int index = Math.abs(arr[i]) - 1;

if (arr[index] < 0)

repeating = Math.abs(arr[i]);

else

arr[index] = -arr[index];

}

for (int i = 0; i < n; i++) {

if (arr[i] > 0) {

missing = i + 1;

break;

}

}

return new int[]{repeating, missing};

}

public static void main(String[] args) {

int[] arr1 = {2, 2};

int[] res1 = findTwoElement(arr1, arr1.length);

System.out.println("Missing: " + res1[1] + ", Repeating: " + res1[0]);

}

}

Problem 2

import java.util.Arrays;

public class Occurences {

static int[] find(int arr[], int n, int x) {

int[] result = {-1, -1};

int first = -1, last = -1;

int low = 0, high = n - 1;

while (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == x) {

first = mid;

high = mid - 1;

} else if (arr[mid] < x) {

low = mid + 1;

} else {

high = mid - 1;

}

}

low = 0;

high = n - 1;

while (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == x) {

last = mid;

low = mid + 1;

} else if (arr[mid] < x) {

low = mid + 1;

} else {

high = mid - 1;

}

}

result[0] = first;

result[1] = last;

return result;

}

public static void main(String[] args) {

int[] arr2 = {1, 3, 5, 5, 5, 5, 67, 123, 125};

int[] res2 = find(arr2, arr2.length, 5);

System.out.println("First Occurrence: " + res2[0] + ", Last Occurrence: " + res2[1]);

}

}

Problem 3

import java.util.Arrays;

public class RearrangeArray {

static void rearrange(int arr[], int n) {

int maxIdx = n - 1, minIdx = 0;

int maxElem = arr[maxIdx] + 1;

for (int i = 0; i < n; i++) {

if (i % 2 == 0) {

arr[i] += (arr[maxIdx] % maxElem) \* maxElem;

maxIdx--;

} else {

arr[i] += (arr[minIdx] % maxElem) \* maxElem;

minIdx++;

}

}

for (int i = 0; i < n; i++) {

arr[i] = arr[i] / maxElem;

}

}

public static void main(String[] args) {

int[] arr3 = {1, 2, 3, 4, 5, 6};

rearrange(arr3, arr3.length);

System.out.println("Rearranged Array: " + Arrays.toString(arr3));

}

}

Problem 4

public class KElement {

static int kthElement(int arr1[], int arr2[], int n, int m, int k) {

if (n > m) return kthElement(arr2, arr1, m, n, k);

if (n == 0) return arr2[k - 1];

if (k == 1) return Math.min(arr1[0], arr2[0]);

int i = Math.min(n, k / 2), j = Math.min(m, k / 2);

if (arr1[i - 1] > arr2[j - 1]) {

return kthElement(arr1, Arrays.copyOfRange(arr2, j, m), n, m - j, k - j);

} else {

return kthElement(Arrays.copyOfRange(arr1, i, n), arr2, n - i, m, k - i);

}

}

public static void main(String[] args) {

int[] arr4\_1 = {2, 3, 6, 7, 9};

int[] arr4\_2 = {1, 4, 8, 10};

int res4\_1 = kthElement(arr4\_1, arr4\_2, arr4\_1.length, arr4\_2.length, 5);

System.out.println("Kth Element: " + res4\_1);

}

}