**Practical 1: Implement Smoothing by Binning**

import java.util.\*;

class DataBinning {

public static void main(String[] args) {

int binLen = 3;

String inputString2 = "4 8 15 21 21 24 25 28 34";

String[] numArrStr = inputString2.split(" ");

int[] arr = new int[numArrStr.length];

for (int i = 0; i < arr.length; i++) {

arr[i] = Integer.parseInt(numArrStr[i]);

}

Arrays.sort(arr);

int binCount = arr.length/binLen;

int[][] bins = new int[binCount][binLen];

for (int i = 0; i < bins.length; i++) {

for (int j = 0; j < bins[i].length; j++) {

bins[i][j] = arr[i\*binLen + j];

}

}

System.out.println("The bins are:");

display(bins);

getMeanBins(bins);

getMedianBins(bins);

getBoundryBins(bins);

}

static void display(int[][] arr) {

System.out.println("[");

for (int[] bin : arr) {

System.out.println(Arrays.toString(bin));

}

System.out.println("]");

}

static void getMeanBins(int[][] bins) {

int[][] meanBins = new int[bins.length][bins[0].length];

int mean;

for (int i = 0; i < meanBins.length; i++) {

// Calculate mean

mean = 0;

for (int value : bins[i]) {

mean += value;

}

mean = mean / bins[i].length;

// Change the value to mean in meanNins

for (int j = 0; j < meanBins.length; j++) {

meanBins[i][j] = mean;

}

}

System.out.println("The mean bins are:");

display(meanBins);

}

static void getMedianBins(int[][] bins) {

int[][] medianBins = new int[bins.length][bins[0].length];

int median;

for (int i = 0; i < medianBins.length; i++) {

median = bins[i].length%2==0?(bins[i][(int)(bins[i].length/2)] + bins[i][(int)(bins[i].length/2) - 1])/2:bins[i][(int)(bins[i].length/2)];

for (int j = 0; j < medianBins.length; j++) {

medianBins[i][j] = median;

}

}

System.out.println("The median bins are:");

display(medianBins);

}

static void getBoundryBins(int[][] bins) {

int[][] boundryBins = new int[bins.length][bins[0].length];

int minVal;

int maxVal;

for (int i = 0; i < boundryBins.length; i++) {

minVal = bins[i][0];

maxVal = bins[i][bins[i].length-1];

for (int j = 0; j < boundryBins.length; j++) {

boundryBins[i][j] = (bins[i][j] - minVal)<(maxVal - bins[i][j]) ? minVal : maxVal;

}

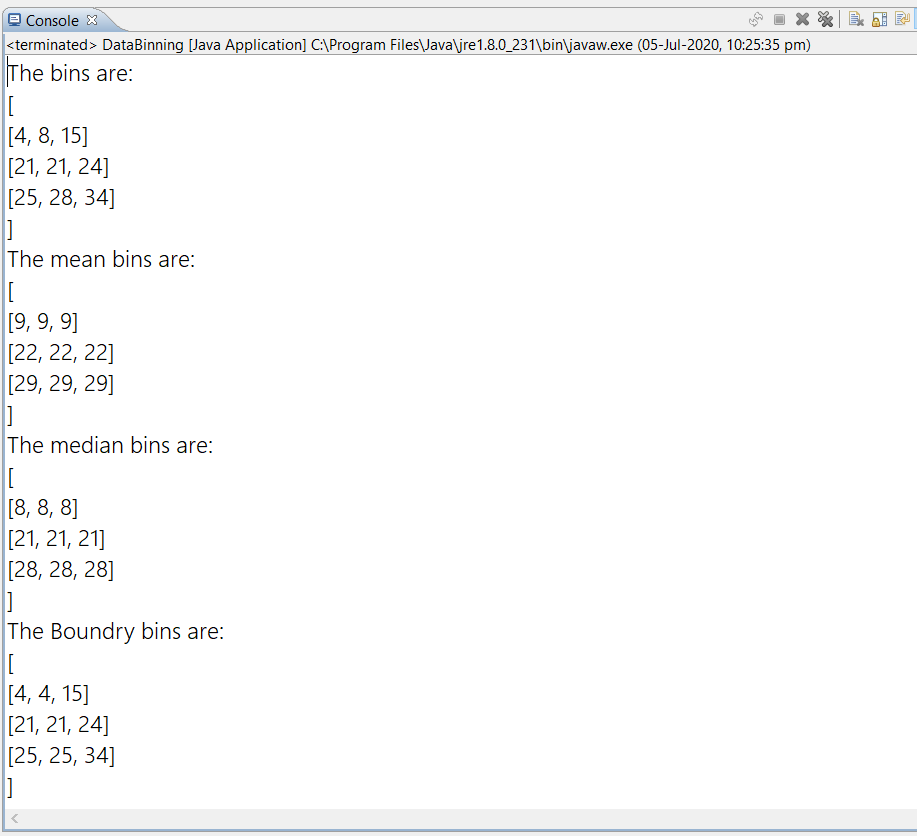
}

System.out.println("The Boundry bins are:");

display(boundryBins);

}

}

**Output:**