//zadanie1  
  
 public static int[] revert(int[] arr) {  
  
 int[] tab = new int[arr.length];  
  
 for (int i = 0; i < arr.length; i++) {  
 arr[arr.length - i - 1] = tab[i];  
  
 }  
 return tab;  
 }  
  
 // zadanie2  
  
 public static String arrayToString(int[] arr){  
 String napis="\"";  
  
 for (int i = 0; i <arr.length ; i++) {  
 if(i == (arr.length - 1)){  
 napis+=arr[i]+"\"";  
 continue;  
 }  
  
 napis+=arr[i]+",";  
  
  
  
 }  
  
 return napis;  
 }  
  
 //zadanie3  
 public static int[] getEven(int[] arr){  
 int []tab=new int[*getNumberOfEven*(arr)];  
 int count=0;  
 for (int i = 0; i <arr.length ; i++) {  
 if (arr[i]%2==0){  
 tab[count]=arr[i];  
 count++;  
 }  
  
 }  
  
 return tab;  
 }  
  
 public static int getNumberOfEven(int[] arr){  
 int count=0;  
 for (int i = 0; i <arr.length ; i++) {  
 if (arr[i]%2==0){  
 count++;  
 }  
 }  
 return count;  
 }  
   
   
 // zadanie4  
 public static boolean arrayEquals(int[] arr, int[] arr2){  
 boolean isEquals=true;  
 if(arr.length==arr2.length){  
 for (int i = 0; i <arr.length ; i++) {  
 if (arr[i]!=arr2[i]){  
 isEquals=false;  
 break;  
 }  
 }  
  
 }else  
 isEquals=false;  
  
   
 return isEquals;  
 }  
  
  
 // zadanie5  
  
 public static void printArray(int[] arr){  
  
 for (int i = 0; i <arr.length ; i++) {  
 System.*out*.println(arr[i]);  
 }  
  
 }  
// zadanie 6  
 public static boolean isInArray(double[] arr, double num){  
 boolean isInArray=false;  
  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i]==num){  
 isInArray=true;  
 break;  
 }  
  
 }  
  
 return isInArray;  
 }  
  
  
 // zadanie 7  
 public static double getAlgebraicMean(double[] arr){  
 double srednia = 0;  
  
 for (int i = 0; i < arr.length; i++) {  
 srednia+=arr[i];  
  
 }  
 srednia=srednia/(arr.length-1);  
  
  
 return srednia;  
 }  
  
 //zadanie 8  
  
 public static boolean isIncreasing(int[] arr){  
 boolean isIncrasing=true;  
 for (int i = 1; i <arr.length ; i++) {  
 if(i!=arr.length-1){  
 if (arr[i - 1] > arr[i]) {  
 isIncrasing=false;  
 }  
 }  
 }  
  
 return isIncrasing;  
 }