מטלה 3 מערכים –דוד אלקובי

1)

Scanner in = new Scanner(System.*in*);  
int j=9;  
int[] arr=new int[10];  
int[]reveresArr=new int[10];  
for (int i = 0; i <10 ; i++) {  
 arr[i] = in.nextInt();  
 reveresArr[j]=arr[i];  
 j--;

2)

Scanner in = new Scanner(System.*in*);  
int j=6;  
int[] arr=new int[10];  
int[]reveresArr=new int[10];  
for (int i = 0; i <7 ; i++) {  
 arr[i] = in.nextInt();  
 reveresArr[j]=arr[i];  
 j--;  
}  
for (int i = 0; i < 7; i++) {  
 if (reveresArr[i]%2==0)  
 System.*out*.println(reveresArr[i]);

3)

public static void numIndex (int[]arr){  
 for (int i = 0; i < arr.length-1; i++) {  
 if (arr[i]>arr[i+1])  
 System.*out*.println("index: "+i);  
 }  
}  
  
public static void main(String[] args) {  
 Scanner in=new Scanner(System.*in*);  
 int [] arr=new int[8];  
 for (int i = 0; i < 8; i++) {  
 arr[i]=in.nextInt();  
 }  
 *numIndex*(arr);

4)

public static boolean ascendingOrder(int[] arr) {  
 int i = 1;  
 while ((arr[i] > arr[i - 1]) && i < arr.length - 1)  
 i++;  
 if (i == arr.length-1)  
 return true;  
 else  
 return false;

5)

public static void moveUpp (String [] arr){  
 int j=1;  
 String temp="";  
 String [] arr1=new String [arr.length];  
 for (int i = 0; i < arr.length-1; i++) {  
 arr1[j]=arr[i];  
 j++;  
 }  
 arr1[0]=arr[arr.length-1];  
 for (int i = 0; i < arr1.length; i++) {  
 System.*out*.print(arr1[i]+" ");  
 }  
 }  
  
 public static void main(String[] args) {  
 Scanner in=new Scanner(System.*in*);  
 String [] arr=new String[4];  
 for (int i = 0; i < arr.length; i++) {  
 arr[i]=in.nextLine();  
 }  
 *moveUpp*(arr);  
 }  
  
}

6)

public static void main(String[] args) {  
 int j=0;  
 int [] arr=new int[10];  
 int [] arr1=new int[10];  
 Scanner in=new Scanner(System.*in*);  
 for (int i = 0; i < 10; i++) {  
 arr[i]=in.nextInt();  
 if (arr[i]%2==0){  
 arr1[j]=arr[i];  
 j++; }  
 else  
 arr1[10-j]=arr[i];  
 }  
  
   
 }  
}

7)

public static void replace (int[]arr){  
 Scanner in=new Scanner(System.*in*);  
 int j= arr.length-1, i=0, num=0;  
 System.*out*.println("enter num: ");  
 num=in.nextInt();  
 while (num>arr[i]){  
 i++;  
 }  
 while (j>i){  
 arr[j]=arr[j-1];  
 j--;  
 }  
 arr[i]=num;  
 for (int k = 0; k < arr.length; k++) {  
 System.*out*.print(arr[k]+" ");

8)

public static boolean doesIn (int [] arr){  
 int n= arr.length;  
 Scanner in =new Scanner(System.*in*);  
 boolean present []=new boolean[n];  
 for (int i = 0; i < n; i++) {  
 if (arr[i]>=n||arr[i]<0)  
 return false;  
 present [arr[i]]=true;  
 }  
 for (int i = 0; i <n ; i++) {  
 if (!present[i])  
 return false;  
 }  
 return true;  
  
 }

9א)

public static int [] uniteArrays(int [] arr,int [] arr1){  
  
 int j=0;  
 int arrUnited []=new int [arr.length+ arr1.length];  
 for (int i = 0; i < arr.length; i++) {  
 arrUnited[i]=arr[i];  
 }  
 for (int i = arr.length; i < arrUnited.length; i++) {  
 arrUnited[i]=arr1[j];  
 j++;  
 }  
 return arrUnited ;  
}

9ב)

public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 int arr[] = new int[3];  
 for (int i = 0; i < arr.length; i++)  
 arr[i] = in.nextInt();  
  
 int arr1[] = new int[5];  
 for (int i = 0; i < arr1.length; i++)  
 arr1[i] = in.nextInt();  
  
 System.*out*.println(Arrays.*toString*(*uniteArrays*(arr, arr1)));  
  
 }  
}

10)

public static int[] unitedArraysAlternately(int[] arr, int[] arr1) {  
 if (arr.length == arr1.length) {  
 int j = 0, k = 0;  
 int[] unitedArray = new int[arr.length + arr1.length];  
 for (int i = 0; i < unitedArray.length; i += 2) {  
 unitedArray[i] = arr[j];  
 j++;  
 }  
 for (int i = 1; i < unitedArray.length; i += 2) {  
 unitedArray[i] = arr1[k];  
 k++;  
 }  
 return unitedArray;  
 }  
 return new int[0];  
 }  
}

11)

public static void twoArrays(int[] arr) {  
 int k = 0, j = 0, posCnt = 0, negCnt = 0;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] > 0)  
 posCnt++;  
 else negCnt++;  
  
  
 }  
 int posArr[] = new int[posCnt];  
 int negArr[] = new int[negCnt];  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] > 0) {  
 posArr[j] = arr[i];  
 j++;  
 } else {  
 negArr[k] = arr[i];  
 k++;  
 }  
 }  
 System.*out*.println(Arrays.*toString*(posArr));  
 System.*out*.println(Arrays.*toString*(negArr));  
 }  
}

12)

public static void Precipitation(double arr[]) {  
 int daysRaining=0, cntRainySeason=0;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] > 0) {  
 daysRaining++;  
 if (daysRaining >= 3) {  
 cntRainySeason++;  
 daysRaining = 0;  
 }  
 } else  
 daysRaining = 0;  
 }  
 System.*out*.println("cntRainySeason: "+cntRainySeason);  
 double maxRain=0;  
 int maxRainIndex=0;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] > maxRain) {  
 maxRainIndex = i + 1;  
 maxRain = arr[i];  
 }  
 }  
 System.*out*.println("the most raining day was day num: "+maxRainIndex+" and the amount was: "+maxRain);  
  
 }

13)

package arreys;  
  
import java.util.Arrays;  
  
public class q13 {  
 public static int [] merge(int[] arr1, int[] arr2) {  
 int k = 0, j = 0;  
 int arrMerge[] = new int[arr1.length + arr2.length];  
 while (k<arr1.length||j< arr2.length)  
 if ()  
  
 return arrMerge;  
  
 }  
  
 public static void main(String[] args) {  
 int arr[] = {4, 7, 7, 15, 20, 21, 33};  
 int arr1[] = {2, 5, 7, 8, 10, 35};  
 Arrays.*toString*(*merge*(arr, arr1));  
  
 }  
}