

Assignment 2

Use arrays to structure the raw data and to perform data comparison & operations

Write a program which creates an integer array and displays sum of its elements.

```
file.java

1 // Write your code here
2 import java.util.Random;
3 public class file {
4     public static void main(String[] args) {
5         Random rd = new Random();
6         int[] arr = new int[5];
7         int sm = 0;
8         for (int i = 0; i < arr.length; i++) {
9             arr[i] = rd.nextInt();
10            sm += arr[i];
11        }
12        System.out.println(sm);
13    }
14 }
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a program which performs addition of elements which are stored in two arrays of type double.

Arrays lengths may be variable in size. The resultant values must be stored in an integer array. Display the resultant integer array in a formatted way.

Example:

Input:

dInputArray1[]	10.0	20.0	30.0		
dInputArray2[]	20.0	50.0	30.0	70.0	80.0

Output:

iSumArray[]	10	70	60	70	80
-------------	----	----	----	----	----

```
file.java

1 // Write your code here
2 import java.util.Scanner;
3 import java.util.Arrays;
4
5 public class file{
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         int n = input.nextInt();
10        int m = input.nextInt();
11
12        double[] arr1 = new double[n];
13        double[] arr2 = new double[m];
14
15        int k = Math.max(n,m);
16        int[] sumArray = new int[k];
17        Arrays.fill(sumArray, 0);
18
19        for(int i = 0; i < n; i++){
20            arr1[i] = input.nextDouble();
21            sumArray[i] += arr1[i];
22        }
23
24        for(int i = 0; i < m; i++){
25            arr2[i] = input.nextDouble();
26            sumArray[i] += arr2[i];
27        }
28        for(int i = 0; i < k; i++){
29            System.out.println(sumArray[i]);
30        }
31    }
32 }
```

Use arrays to structure the raw data and to perform data comparison & operations

Write a **method** that receives a name as **parameter** and prints on the console.
"Hello, <name>!" Example

Input	Output
Peter	Hello, Peter!

```
file.java

1 // Write your code here
2 import java.util.*;
3
4 public class file{
5     static void greeting(String name){
6         System.out.println("Hello, "+ name + "!");
7     }
8
9     public static void main(String[] args) {
10         Scanner input = new Scanner(System.in);
11         String name = input.nextLine();
12
13         greeting(name);
14
15     }
16 }
```

Use arrays to structure the raw data and to perform data comparison & operations

Create a method **GetMax(int a, int b, int c)**, that returns maximal of three numbers. Write a program that reads three numbers from the console and prints the biggest of them.

Examples

Input	Output
1	3
2	
3	

Input	Output
-100	-100
-101	
-102	

```
file.java

1 // Write your code here
2 import java.util.*;
3
4 public class file{
5     static void GetMax(int a, int b, int c){
6         if(a >= b && a >= c){
7             System.out.println(a);
8         }
9         else if(b >= a && b >= c){
10            System.out.println(b);
11        }
12        else{
13            System.out.println(c);
14        }
15    }
16
17    public static void main(String[] args) {
18        Scanner input = new Scanner(System.in);
19        int a = input.nextInt();
20        int b = input.nextInt();
21        int c = input.nextInt();
22
23        GetMax(a, b, c);
24
25    }
26 }
```

Use arrays to structure the raw data and to perform data comparison & operations

Write a method that **prints the digits** of a given decimal number in a **reversed order**. Examples

Input	Output
256	652

```
file.java

1 // Write your code here
2 import java.util.*;
3
4 public class file{
5     static void PrintReverse(int n){
6         while(n > 0){
7             int lastDigit = n % 10;
8             n /= 10;
9             System.out.print(lastDigit);
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner input = new Scanner(System.in);
15        int a = input.nextInt();
16        PrintReverse(a);
17    }
18 }
19 }
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a Boolean method **IsPrime(n)** that check whether a given integer number **n** is prime.

Examples:

n	IsPrime(n)
0	false
1	false
2	true
3	true
4	false
5	true
323	false
337	true
6737626471	true
117342557809	false

```
file.java

1 // Write your code here
2 import java.util.*;
3
4 public class file{
5     static boolean IsPrime(int n){
6         if(n == 0 || n == 1){
7             return false;
8         }
9
10        if(n < 4){
11            return true;
12        }
13
14        if(n % 2 == 0 || n % 3 == 0)
15            return false;
16
17        for(int i = 5; i <= Math.sqrt(n); i+=6){
18            if(n % i == 0 || n % (i+2) == 0){
19                return false;
20            }
21        }
22
23        return true;
24    }
25
26    public static void main(String[] args) {
27        Scanner input = new Scanner(System.in);
28        int a = input.nextInt();
29        System.out.println(IsPrime(a));
30
31    }
32 }
```




Use arrays to structure the raw data and to perform data comparison & operations

Write a method that calculates **all prime numbers in given range** and returns them as list of integers

Write a method to **print a list of integers**. Write a program that takes two integer numbers (each at a separate line) and prints all primes in their range, separated by a comma.

Examples

Start and End Number	Output
0 10	2, 3, 5, 7
5 11	5, 7, 11
100 200	101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199
250 950	251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947
100 50	<i>(empty list)</i>

```
file.java

1
2 //start coding a new Program
3
4 import java.util.*;
5
6 public class file{
7     private static int j = 0;
8
9     static boolean IsPrime(int n){
10         if(n == 0 || n == 1){
11             return false;
12         }
13
14         if(n < 4){
15             return true;
16         }
17
18         if(n % 2 == 0 || n % 3 == 0)
19             return false;
20
21         for(int i = 5; i <= Math.sqrt(n); i+=6){
22             if(n % i == 0 || n % (i+2) == 0){
23                 return false;
24             }
25         }
26
27         return true;
28     }
29
30     static int[] generatePrimes(int a, int b){
31         int[] primes = new int[1000];
32
33         for(int i = a; i <= b; i++){
34             if(IsPrime(i)){
35                 primes[j++] = i;
36             }
37         }
38
39         return primes;
40     }
41
42     static void printList(int[] arr){
43         for(int i = 0; i < j-1; i++){
44             System.out.print(arr[i] + ", ");
45         }
46         System.out.println(arr[j-1]);
47     }
48
49     public static void main(String[] args) {
50         Scanner input = new Scanner(System.in);
51         int a = input.nextInt();
52         int b = input.nextInt();
53         printList(generatePrimes(a, b));
54     }
55 }
56
57
58
59 //Click Enter to expand window
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a program that can **calculate the area** of **four different geometry figures** - triangle, square, rectangle and circle.

On the first line you will get the **figure type**. Next you will get parameters for the chosen figure, **each on a different line**:

- Triangle - side and height
- Square - side
- Rectangle - width and height
- Circle - radius

The output should be rounded to the second digit after the decimal point: **Examples**

Input	Output
triangle 3 6	9.00
rectangle 4 5	20.00

```
file.java

1 //start coding a new Program
2
3 import java.util.*;
4
5 public class file{
6     public static double area(String figureType){
7         Scanner input = new Scanner(System.in);
8         double ans = 0;
9         int side, height, radius, width;
10        switch(figureType){
11            case "triangle":
12                side = input.nextInt();
13                height = input.nextInt();
14                ans = (side * height)/2.0;
15                break;
16            case "square":
17                side = input.nextInt();
18                ans = side * side;
19                break;
20            case "rectangle":
21                width = input.nextInt();
22                height = input.nextInt();
23                ans = width * height;
24                break;
25            case "circle" :
26                radius = input.nextInt();
27                ans = 3.14*radius*radius;
28                break;
29            default:
30                System.out.println("Enter the correct figure type:");
31        }
32
33        return ans;
34    }
35
36    public static void main(String[] args) {
37        Scanner input = new Scanner(System.in);
38        String figureType = input.next();
39        System.out.println(area(figureType));
40    }
41 }
42 }
43 }
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a method which accepts two integer arrays and returns an array of unique elements.

Example:

Array 1 = { 10, 5, 20, 15, 25, 30}

Array 2 = {50, 12, 5, 30, 15, 70}

Result_Array = {10, 20, 25, 50, 12, 70}

```
Int [] uniqElements(int array1[], int array2[]);
```

```
file.java

1 //start coding a new Program
2
3 import java.util.*;
4
5 public class file{
6     public static int[] uniqElements(int[] arr1, int[] arr2){
7         Set<Integer> data = new LinkedHashSet<Integer>();
8         boolean flag;
9         for(int i: arr1){
10             flag = true;
11             for(int j: arr2){
12                 if(i == j){
13                     flag = false;
14                 }
15             }
16             if(flag)
17                 data.add(i);
18         }
19         for(int i: arr2){
20             flag = true;
21             for(int j: arr1){
22                 if(i == j){
23                     flag = false;
24                 }
25             }
26             if(flag)
27                 data.add(i);
28         }
29         int[] ans = new int[data.size()];
30         int j = 0;
31         for(int i: data){
32             ans[j++] = i;
33         }
34         return ans;
35     }
36 }
37
38
39 public static void main(String[] args) {
40     int[] arr1 = {10, 5, 20, 15, 25, 30};
41     int[] arr2 = {50, 12, 5, 30, 15, 70};
42     int[] ans = uniqElements(arr1, arr2);
43     for(int i :ans){
44         System.out.print(i + ", ");
45     }
46 }
47 }
48 }
49
50
51
52 //Click Enter to expand window
```



Use arrays to structure the raw data and to perform data comparison & operations

Analyze below given code and predict the output.

```
public class TwoDArray
{
    //-----
    //  Creates a 2D array of integers, fills it with increasing
    //  integer values, then prints them out.
    //-----
    public static void main (String[] args)
    {
        int[][] table = new int[5][10];
        // Load the table with values
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                table[row][col] = row * 10 + col;
        // Print the table
        for (int row=0; row < table.length; row++)
        {
            for (int col=0; col < table[row].length; col++)
                System.out.print (table[row][col] + "\t");
            System.out.println();
        }
    }
}
```

output

1	0	1	2	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16	17	18	19
3	20	21	22	23	24	25	26	27	28	29
4	30	31	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47	48	49



Use arrays to structure the raw data and to perform data comparison & operations

Write a method which accepts two matrices of Size N X N and returns summation of resultant Matrix.

Example:

Matrix A: [1,2,3] [4,5,6]

Matrix B: [4,5,6] [7,8,9]

Matrix C = A + B = [5,7,9] [11,13,15]


```
Main.java

1 import java.util.*;
2
3 public class Main{
4
5     public static int[][] matrixSum(int[][] a, int[][]b, int n){
6         int[][] c = new int[n][n];
7         for(int i = 0; i < n; i++){
8             for(int j = 0; j < n; j++){
9                 c[i][j] = a[i][j] + b[i][j];
10            }
11        }
12
13        return c;
14    }
15
16    public static void main(String[] args) {
17        Scanner input = new Scanner(System.in);
18        int n = input.nextInt();
19
20        int[][] a = new int[n][n];
21        int[][] b = new int[n][n];
22
23        for(int i = 0; i < n; i++){
24            for(int j = 0; j < n; j++){
25                a[i][j] = input.nextInt();
26            }
27        }
28        for(int i = 0; i < n; i++){
29            for(int j = 0; j < n; j++){
30                b[i][j] = input.nextInt();
31            }
32        }
33
34        int[][] res = matrixSum(a, b, n);
35
36        for(int i = 0; i < n; i++){
37            for(int j = 0; j < n; j++){
38                System.out.print(res[i][j] + " ");
39            }
40            System.out.println();
41        }
42    }
43 }
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a method `public static boolean isRowMagic(int[][] a)` that checks if the array is row-magic (this means that every row has the same row sum).

```
file.java

1 import java.util.*;
2
3 public class file{
4
5     public static boolean isRowMagic(int[][] a, int n, int m){
6         int[] allRowSum = new int[n];
7         int k = 0;
8         for(int i = 0; i < n; i++){
9             int rowSum = 0;
10            for(int j = 0; j < m; j++){
11                rowSum += a[i][j];
12            }
13            allRowSum[i] = rowSum;
14        }
15
16        for(int i = 0; i < n-1; i++){
17            if(allRowSum[i] != allRowSum[i+1]){
18                return false;
19            }
20        }
21        return true;
22    }
23
24    public static void main(String[] args) {
25        Scanner input = new Scanner(System.in);
26        int n = input.nextInt();
27        int m = input.nextInt();
28
29        int[][] a = new int[n][m];
30
31
32        for(int i = 0; i < n; i++){
33            for(int j = 0; j < m; j++){
34                a[i][j] = input.nextInt();
35            }
36        }
37
38        System.out.println(isRowMagic(a, n, m));
39
40    }
41 }
42
43
44
45 //Click Enter to expand window
```



Use arrays to structure the raw data and to perform data comparison & operations

Write a method public
static boolean
isMagic(int[][] a)

that checks if the array is a magic square. This means that it must be square, and that all row sums, all column sums, and the two diagonal-sums must all be equal.

```
file.java

1 import java.util.*;
2
3 public class file{
4
5     public static boolean isRowMagic(int[][] a, int n){
6         int[] allRowSum = new int[n];
7         int[] allColSum = new int[n];
8         int diagonal1 = 0, diagonal2 = 0;
9
10        for(int i = 0; i < n; i++){
11            int rowSum = 0;
12            int colSum = 0;
13            for(int j = 0; j < n; j++){
14                rowSum += a[i][j];
15                colSum += a[j][i];
16            }
17            allRowSum[i] = rowSum;
18            allColSum[i] = colSum;
19        }
20
21        for(int i = 0; i < n; i++){
22
23            for(int j = 0; j < n; j++){
24                if(i == j){
25                    diagonal1 += a[i][j];
26                }
27            }
28            diagonal2 += a[i][n-1-i];
29        }
30
31        // checking row magic
32        for(int i = 0; i < n-1; i++){
33            if(allRowSum[i] != allRowSum[i+1] || allColSum[i] != allColSum[i+1]){
34                return false;
35            }
36        }
37        if(diagonal1 != diagonal2){
38            return false;
39        }
40        return true;
41    }
42
43    public static void main(String[] args) {
44        Scanner input = new Scanner(System.in);
45        int n = input.nextInt();
46
47        int[][] a = new int[n][n];
48
49
50        for(int i = 0; i < n; i++){
51            for(int j = 0; j < n; j++){
52                a[i][j] = input.nextInt();
53            }
54        }
55
56        System.out.println(isRowMagic(a, n));
57
58    }
59 }
60
```