**Exercise 1**

public static int arrayAverage(int[] array) {

int average = 0, sum = 0;

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

sum = sum + array[i];

}

average = sum/array.length;

return average;

}

**Exercise 2**

public static boolean rowsEqTargetSum(int targetSum, int[][] a) {

int row, col, sum;

for(row=0; row < square.length; row++) {

sum=0;

for(col=0; col<n; col++) {

int value = a[row][col];

sum += value;

}

System.out.println(" = "+sum);

if(sum != targetSum) {

System.out.println("Row sum incorrect : Not a magic Square !");

return false;

}

return true;

}

}

**Exercise 3**

public static boolean diagonalEqTargetSum(int targetSum, int[][] a) {

int row, col, sum;

final int n=a.length;

sum=0;

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

row = n-1 - pos;

col = pos;

int value = a[row][col];

sum += value;

}

if(sum != targetSum) {

System.out.println("Diagonal is incorrect : Not a magic Square !");

return false;

}

return true;

}

**Exercise 4**

public static boolean allNumbersRepresented(int[][] a) {

int row, col;

final int n=a.length;

final int nSquare=n\*n;

boolean[] flag= new boolean[n\*n];

for(row=0; row<n; row++){

for(col=0; col<n; col++){

int num = a[row][col];

if (n < 1 || num > nSquare){

System.out.println("Number out of range : Not a magic Square !");

return false;

}

if (flag[num-1]) {

System.out.println("Duplicate number : Not a magic Square !");

return false;

}

flag[num-1] = true;

}

}

return true;

}

public static void main(String []args){

int[][] a ={{4,9,2},

{3,5,7},

{8,1,6}};

final int n=a.length;

final int targetSum=n\*(n\*n+1)/2;

if (!rowsEqTargetSum(targetSum, a)){

return;

}

if (!diagonalEqTargetSum(targetSum, a)){

return;

}

if (!allNumbersRepresented(a)){

return;

}

System.out.println(" The following two dimensional array is Magic !");

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

System.out.println(Arrays.toString(a[i]));

}

}