package lab8;

/\*\*

\*

\* @author gawitt

\*/

public class Lab8 {

int [][] matrix = new int[4][4]; // instantiated a multi-dim. array

public void populateMatrix(){

System.out.println("Populate the Matrix....");//loading

int [] num = new int[50]; // created an array with 50 positions

//load the array.

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

num[i] = i+1; //adding 1 because of 1-50 condition

System.out.print(num[i]+" ");//

}//for

System.out.println("");

//shuffle

System.out.println("Shuffling....");

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

int index = (int)(Math.random() \* num.length);

//swap

int temp = num[i];

num[i] = num[index];

num[index] = temp;

//System.out.println("Swapping " + num[i] + " with "+ num[index]);//

}//for

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

System.out.print(num[i] + " ");//

}//for

System.out.println("");//

//load the 2D array

int k = 0;

for (int i = 0; i < matrix.length; i++) {//rows loading

for (int j = 0; j < matrix[i].length; j++) { //columns loading

matrix[i][j] = num[k]; // loading the previous shuffled

k++;

}//j

}//i

}//populateMatrix

public void outputMatrix(){

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

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

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

System.out.print(matrix[i][j] + "\t");

}//i

System.out.println("");

}//j

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}//outputMatrix

public int[] findLocationOflargest(){ //returns two numbers to indicate the location of the array

System.out.println("Finding Largest...."); //echo

int largest = -1; //initial condition

int[] location = {-1,-1}; // intial condition array to reveal the location

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

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

if(largest < matrix[i][j]){ // checking

largest = matrix[i][j]; // declaring

location[0] = i; // change the values to reflect location

location[1] = j; // changes values to refelect ythe location

System.out.println("New largest is " + largest);

}//if

}//j

}//i

return location;

}//findLargest

public int[] getDiagonal(){//returning array

System.out.println("Getting diagonal....");

int[] diagonal = {-1,-1,-1,-1}; //loaded with its own default

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

diagonal[i] = matrix[i][i];

}//for

return diagonal; //no casting needed because the diagonal

}//diagonal

public void transpose(){ //no return

System.out.println("Transposing the array ....");

int[][] matrixT = new int[4][4];

for (int i = 0; i < 4; i++) { // 4 is the number of times it must run the rows

for (int j = 0; j < 4; j++) { // 4 is the number of times it must run the columns

matrixT[j][i] = matrix[i][j];

}// for j

}//for i

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

System.out.print(matrixT[j][i] + "\t");

}// i

System.out.println("");

}//j

}// transpose method

public int[] find(int key){ // argument of an integer method and value to be found

System.out.println("searching for the number " + key);

int[] location = {-1,-1}; // starting default

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

if(matrix[i][j] == key){ // test searching value

System.out.println("found it!");

location[0] = i;

location[1] = j;

break; // ending the find method and simply printing out the found array

}//if

}//for j

}//for i

return location; // returning the values of the array

}//find method

public static void main(String[] args) {

Lab8 ans = new Lab8(); // instantiating an array to play with

ans.populateMatrix();//adding values to the matrix

ans.outputMatrix(); // testing

int[] place = new int[2];

//making a single dimensional array to hold the values which we are using as place holders

place = ans.findLocationOflargest(); // finds place of the largest in the istantiated array ans

System.out.println("The location of the largest in the two dimensional array:("+ place[0]+ "," + place[1]+")");

System.out.println("The largest is: " + ans.matrix[place[0]][place[1]] );// matrix is 2D array

//================================

//work from 4/23/17

//

int[] diagonal = new int[4];

diagonal = ans.getDiagonal();

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

System.out.print(diagonal[i]+ "\t");

}//for

System.out.println("");

//transpose method test

ans.transpose();

//find method test

place = ans.find(14);

if(place[0] == -1){ //not there

System.out.println("didnt find it in the matrix.");

}//if

else{

System.out.println("The location is at:("+ place[0]+ "," + place[1]+")");

}//else

}//main

}//class

run:

Populate the Matrix....

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Shuffling....

47 5 3 21 45 28 4 50 12 14 27 29 32 36 13 19 17 48 6 49 37 39 31 43 34 33 11 9 44 2 26 1 10 42 25 40 18 41 23 22 24 16 35 30 15 20 8 38 46 7

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

47 5 3 21

45 28 4 50

12 14 27 29

32 36 13 19

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Finding Largest....

New largest is 47

New largest is 50

The location of the largest in the two dimensional array:(1,3)

The largest is: 50

Getting diagonal....

47 28 27 19

Transposing the array ....

47 5 3 21

45 28 4 50

12 14 27 29

32 36 13 19

searching for the number 14

found it!

The location is at:(2,1)

BUILD SUCCESSFUL (total time: 0 seconds)