Code : 1  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class A\_2d\_Arrays\_Demo {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n1 = s.nextInt();

int n2 = s.nextInt();

int[][] a = new int[n1][n2];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = s.nextInt();

}

}

display2dArray(a);

}

private static void display2dArray(int[][] a) {

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

for (int j = 0; j < a[0].length; j++) {

System.out.print(a[i][j] + " ");

}

System.out.println();

}

}

}

Code : 2  
package Topic\_05\_2D\_Arrays;

import java.util.\*;

public class B\_Matrix\_Multiplication {

public static void main(String[] args) {

Scanner scn = new Scanner(System.in);

/\* Matrix 1 Input \*/

int r1 = scn.nextInt();

int c1 = scn.nextInt();

int[][] a = new int[r1][c1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

/\* Matrix 2 Input \*/

int r2 = scn.nextInt();

int c2 = scn.nextInt();

int[][] b = new int[r2][c2];

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

for (int j = 0; j < b[0].length; j++) {

b[i][j] = scn.nextInt();

}

}

if (c1 != r2) {

System.out.println("Invalid input");

return;

}

int[][] prd = new int[r1][c2];

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

for (int j = 0; j < prd[0].length; j++) {

for (int k = 0; k < c1; k++) {

prd[i][j] += a[i][k] \* b[k][j];

}

}

}

display2dArray(prd);

}

private static void display2dArray(int[][] a) {

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

for (int j = 0; j < a[0].length; j++) {

System.out.print(a[i][j] + " ");

}

System.out.println();

}

}

}

Code : 3  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class C\_StateOfWakanda\_1 {

public static void main(String[] args) {

/\* Matrix 1 Input \*/

int a[][] = takeInput();

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

if (i % 2 == 0) {

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

System.out.println(a[j][i]);

}

} else {

for (int j = a.length - 1; j >= 0; j--) {

System.out.println(a[j][i]);

}

}

}

}

private static int[][] takeInput() {

Scanner scn = new Scanner(System.in);

int r1 = scn.nextInt();

int c1 = scn.nextInt();

int[][] a = new int[r1][c1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

return a;

}

}

Code : 4  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class D\_Spiral\_Matrix {

public static void main(String[] args) {

// TODO Auto-generated method stub

int[][] a = takeInput();

int minr = 0, minc = 0, maxr = a.length - 1, maxc = a[0].length - 1;

int totalCount = a.length \* a[0].length;

int count = 0;

while (count != totalCount) {

for (int i = minr, j = minc; i <= maxr && count != totalCount; i++) {

print(a[i][j]);

count++;

}

minc++;

for (int i = maxr, j = minc; j <= maxc && count != totalCount; j++) {

print(a[i][j]);

count++;

}

maxr--;

for (int i = maxr, j = maxc; i >= minr && count != totalCount; i--) {

print(a[i][j]);

count++;

}

maxc--;

for (int i = minr, j = maxc; j >= minc && count != totalCount; j--) {

print(a[i][j]);

count++;

}

minr++;

}

}

private static void print(int i) {

// TODO Auto-generated method stub

System.out.println(i);

}

private static int[][] takeInput() {

Scanner scn = new Scanner(System.in);

int r1 = scn.nextInt();

int c1 = scn.nextInt();

int[][] a = new int[r1][c1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

return a;

}

}

Code : 5  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class E\_ExitPointOfMatrix {

public static void main(String[] args) {

int[][] a = takeInput();

int dir = 0;

int i = 0, j = 0;

while (true) {

dir = (dir + a[i][j]) % 4;

if (dir == 0) {

j++;

} else if (dir == 1) {

i++;

} else if (dir == 2) {

j--;

} else if (dir == 3) {

i--;

}

if (i < 0) {

i++;

break;

} else if (j < 0) {

j++;

break;

} else if (i == a.length) {

i--;

break;

} else if (j == a[0].length) {

j--;

break;

}

}

System.out.println(i);

System.out.println(j);

}

private static int[][] takeInput() {

Scanner scn = new Scanner(System.in);

int r1 = scn.nextInt();

int c1 = scn.nextInt();

int[][] a = new int[r1][c1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

return a;

}

}

Code : 6  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class F\_RotateBy90Degree {

public static void main(String[] args) throws Exception {

// write your code here

int[][] a = takeInput();

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

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

int temp = a[i][j];

a[i][j] = a[j][i];

a[j][i] = temp;

}

}

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

int li = 0;

int ri = a[0].length - 1;

while (li < ri) {

int temp = a[i][li];

a[i][li] = a[i][ri];

a[i][ri] = temp;

li++;

ri--;

}

}

display(a);

}

public static void display(int[][] arr) {

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

for (int j = 0; j < arr[0].length; j++) {

System.out.print(arr[i][j] + " ");

}

System.out.println();

}

}

private static int[][] takeInput() {

Scanner scn = new Scanner(System.in);

int r1 = scn.nextInt();

// int c1 = scn.nextInt();

int[][] a = new int[r1][r1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

return a;

}

}

Code : 7  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class G\_RingRotate {

public static void main(String[] args) throws Exception {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int m = sc.nextInt();

int[][] arr = new int[n][m];

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

for (int j = 0; j < arr[0].length; j++) {

arr[i][j] = sc.nextInt();

}

}

// Which shell to rotate?

int s = sc.nextInt();

// How many time?

int r = sc.nextInt();

rotateShell(arr, s, r);

display(arr);

sc.close();

}

public static void rotateShell(int[][] arr, int s, int r) {

// Extract OneD Array

int[] oned = fillOnedFromShell(arr, s);

rotate(oned, r);

fillShellFromOned(arr, s, oned);

}

public static int[] fillOnedFromShell(int[][] arr, int s) {

// if Shell = 3, then Box TopLeft = S-1, S-1, Box BottomDown = arr.length-S

int minRow = s - 1;

int minCol = s - 1;

int maxRow = arr.length - s;

int maxCol = arr[0].length - s;

// corners are getting repetitive so -4

// leftWall + bottomWall + rightWall + topWall

// int size = (2 \* (maxRow - minRow + 1)) + (2 \* (maxCol - minCol + 1)) - 4;

int size = 2 \* (maxRow - minRow + maxCol - minCol);

int[] oned = new int[size];

// leftWall

int idx = 0;

for (int i = minRow, j = minCol; i <= maxRow; i++) {

oned[idx] = arr[i][j];

idx++;

}

// bottomWall

for (int i = maxRow, j = minCol + 1; j <= maxCol; j++) {

oned[idx] = arr[i][j];

idx++;

}

// rightWall

for (int i = maxRow - 1, j = maxCol; i >= minRow; i--) {

oned[idx] = arr[i][j];

idx++;

}

// topWall

for (int i = minRow, j = maxCol - 1; j >= minCol + 1; j--) {

oned[idx] = arr[i][j];

idx++;

}

return oned;

}

public static void rotate(int[] oned, int r) {

r = r % oned.length;

if (r < 0)

r += oned.length;

reverse(oned, 0, oned.length - r - 1);

reverse(oned, oned.length - r, oned.length - 1);

reverse(oned, 0, oned.length - 1);

}

public static void reverse(int[] oned, int li, int ri) {

while (li < ri) {

int temp = oned[li];

oned[li] = oned[ri];

oned[ri] = temp;

li++;

ri--;

}

}

public static void fillShellFromOned(int[][] arr, int s, int[] oned) {

// if Shell = 3, then Box TopLeft = S-1, S-1, Box BottomDown = arr.length-S

int minRow = s - 1;

int minCol = s - 1;

int maxRow = arr.length - s;

int maxCol = arr[0].length - s;

// leftWall

int idx = 0;

for (int i = minRow, j = minCol; i <= maxRow; i++) {

arr[i][j] = oned[idx];

idx++;

}

// bottomWall

for (int i = maxRow, j = minCol + 1; j <= maxCol; j++) {

arr[i][j] = oned[idx];

idx++;

}

// rightWall

for (int i = maxRow - 1, j = maxCol; i >= minRow; i--) {

arr[i][j] = oned[idx];

idx++;

}

// topWall

for (int i = minRow, j = maxCol - 1; j >= minCol + 1; j--) {

arr[i][j] = oned[idx];

idx++;

}

}

public static void display(int[][] arr) {

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

for (int j = 0; j < arr[0].length; j++) {

System.out.print(arr[i][j] + " ");

}

System.out.println();

}

}

}

Code : 8  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class H\_StateOfWakanda\_2 {

public static void main(String[] args) {

/\* Matrix 1 Input \*/

int a[][] = takeInput();

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

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

System.out.println(a[i][j]);

}

}

}

private static int[][] takeInput() {

Scanner scn = new Scanner(System.in);

int r1 = scn.nextInt();

// int c1 = scn.nextInt();

int[][] a = new int[r1][r1];

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

for (int j = 0; j < a[0].length; j++) {

a[i][j] = scn.nextInt();

}

}

return a;

}

}

Code : 9  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class I\_SaddlePrice {

public static void main(String[] args) throws Exception {

// write your code here

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int ar[][] = new int[n][n];

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

for (int j = 0; j < ar[0].length; j++) {

ar[i][j] = sc.nextInt();

}

}

//traversing row by row and finding the least element column index and then checking in that

//column if there is any element greater than this or not.

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

int lci = 0;

for (int j = 1; j < ar[0].length; j++) {

if (ar[i][j] < ar[i][lci]) {

lci = j;

}

}

boolean flag = true;

for (int k = 0; k < ar.length; k++) {

if (ar[k][lci] > ar[i][lci]) {

flag = false;

break;

}

}

if (flag == true) {

System.out.println(ar[i][lci]);

return;

}

}

System.out.println("Invalid input");

}

}

Code : 10  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class J\_SearchInASorted2dArray {

public static void main(String[] args) throws Exception {

// write your code here

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int ar[][] = new int[n][n];

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

for (int j = 0; j < ar[0].length; j++) {

ar[i][j] = sc.nextInt();

}

}

int x = sc.nextInt();

int a = 0;

int b = ar[0].length - 1;

while (a < ar.length && b >= 0) {

if (ar[a][b] == x) {

System.out.println(a);

System.out.println(b);

return;

} else if (ar[a][b] > x) {

b--;

} else {

a++;

}

}

System.out.println("Not Found");

}

}

Code : 11  
package Topic\_05\_2D\_Arrays;

import java.util.Scanner;

public class L\_Fibonacci\_DP {

public static void main(String[] args) throws Exception {

// write your code here

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

// System.out.println(fib(n));

System.out.println(fiboMemorize(n, new int[n + 1]));

}

public static int fib(int n) {

if (n <= 1) {

return n;

}

return fib(n - 1) + fib(n - 2);

}

public static int fiboMemorize(int n, int dp[]) {

if (n == 0 || n == 1) {

return n;

}

if (dp[n] != 0) {

return dp[n];

}

int ans = fiboMemorize(n - 1, dp) + fiboMemorize(n - 2, dp);

dp[n] = ans;

return ans;

}

}