Assingment-1

1.

import java.util.Scanner;

public class RightTrianglePattern {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int n = input.nextInt();

input.close();

for (int i = 1; i <= n; i++) {

// print spaces

for (int j = 1; j <= n-i; j++) {

System.out.print(" ");

}

// print stars

for (int j = 1; j <= i; j++) {

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

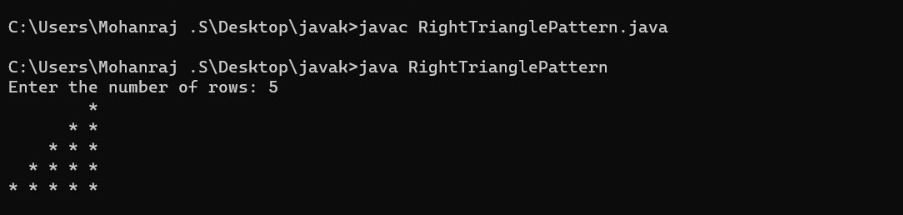
}

System.out.println();

}

}

}



2.

import java.util.Scanner;

public class PascalTriangle {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int n = input.nextInt();

input.close();

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

// print spaces

for (int j = 1; j <= n-i; j++) {

System.out.print(" ");

}

// print numbers

int num = 1;

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

System.out.printf("%4d", num);

num = num \* (i - j) / (j + 1);

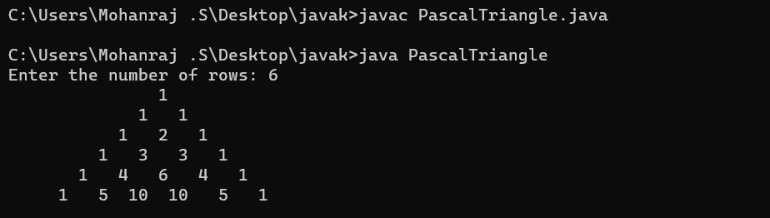
}

System.out.println();

}

}

}



3.

import java.util.Scanner;

public class RectanglePattern {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the symbol: ");

char symbol = input.next().charAt(0);

System.out.print("Enter the number of rows: ");

int rows = input.nextInt();

System.out.print("Enter the number of columns: ");

int columns = input.nextInt();

input.close();

// print the pattern

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= columns; j++) {

System.out.print(symbol + " ");

}

System.out.println();

}

}

}



4.

import java.util.Scanner;

public class NumberPattern {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the number to be printed: ");

int num = input.nextInt();

System.out.print("Enter the maximum number of times to be printed: ");

int max = input.nextInt();

input.close();

// print the pattern

for (int i = 1; i <= max; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(num);

}

System.out.println();

}

for (int i = max-1; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

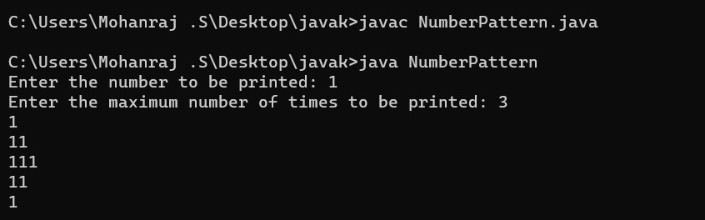
System.out.print(num);

}

System.out.println();

}

}

}

5.

import java.util.Scanner;

public class InvertedFullPyramid {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int rows = input.nextInt();

input.close();

// print the pattern

for (int i = rows; i >= 1; i--) {

// print spaces

for (int j = 1; j <= rows-i; j++) {

System.out.print(" ");

}

// print stars

for (int j = 1; j <= 2\*i-1; j++) {

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

}

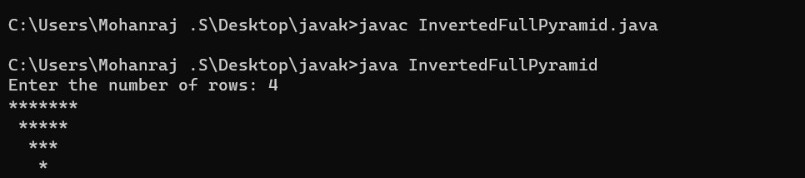
// move to next line

System.out.println();

}

}

}



6.import java.util.Scanner;

public class CharacterPattern {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the character to be printed: ");

char ch = input.next().charAt(0);

System.out.print("Enter the maximum number of times to be printed: ");

int max = input.nextInt();

input.close();

// print the pattern

for (int i = 1; i <= max; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(ch + " ");

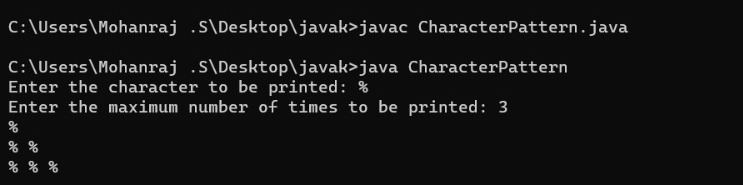
}

System.out.println();

}

}

}



7.import java.util.Scanner;

public class HollowSquarePattern {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the symbol to be printed: ");

char symbol = input.next().charAt(0);

System.out.print("Enter the size of the square: ");

int size = input.nextInt();

input.close();

// print the pattern

for (int i = 1; i <= size; i++) {

for (int j = 1; j <= size; j++) {

// print the symbol for the first and last row

// or for the first and last column

// or for the cells that form the diagonal line

if (i == 1 || i == size || j == 1 || j == size || i == j || j == size-i+1) {

System.out.print(symbol + " ");

} else {

System.out.print(" ");

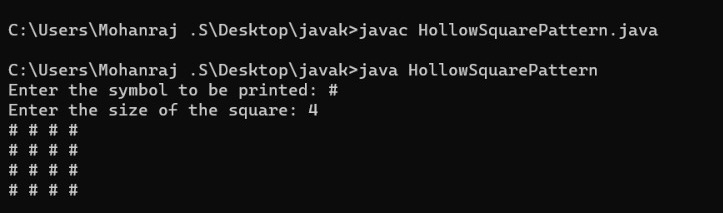
}

}

System.out.println();

}

}

}

8.

public class NumberPattern {

public static void main(String[] args) {

int n = 4;

// loop to print the pattern

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(i + " ");

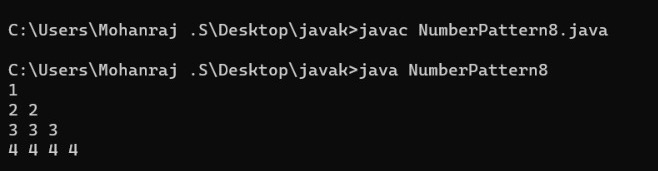
}

System.out.println();

}

}

}



9.public class NumberPattern {

public static void main(String[] args) {

int n = 4;

// loop to print the pattern

int k = 1;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(k \* k + " ");

k++;

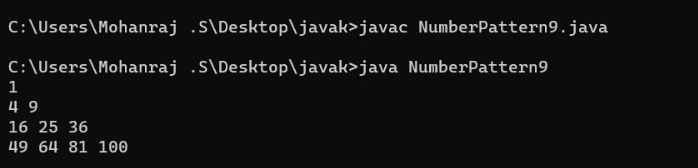
}

System.out.println();

}

}

}



10.public class NumberPattern {

public static void main(String[] args) {

int n = 4;

// print upper half of the pattern

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(i + " ");

}

System.out.println();

}

// print lower half of the pattern

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

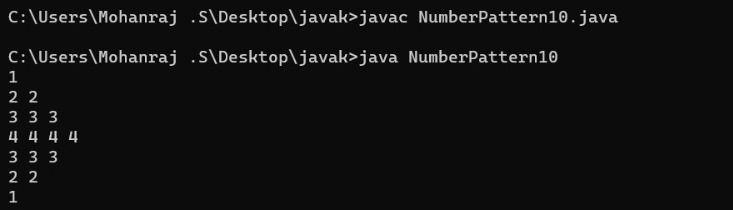
System.out.print(i + " ");

}

System.out.println();

}

}

}

11.import java.util.Scanner;

public class HollowSquareDollarPattern {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the square: ");

int size = sc.nextInt();

for(int i=1; i<=size; i++) {

for(int j=1; j<=size; j++) {

if(i==1 || i==size || j==1 || j==size) {

System.out.print("$");

}

else {

System.out.print(" ");

}

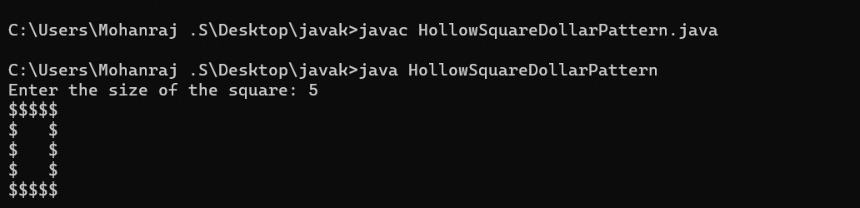
}

System.out.println();

}

sc.close();

}

}

12.

import java.util.Scanner;

public class InvertedPyramidPattern {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int rows = sc.nextInt();

sc.close();

for (int i = rows; i >= 1; i--) {

for (int j = 1; j <= rows - i; j++) {

System.out.print(" ");

}

for (int j = 1; j <= 2 \* i - 1; j++) {

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

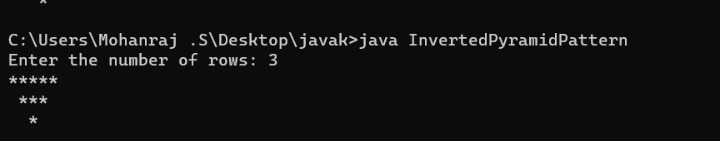
}

System.out.println();

}

}

}



13.public class MatrixMultiplication {

public static void main(String[] args) {

int[][] mat1 = {{1, 2}, {5, 3}}; // input matrix 1

int[][] mat2 = {{2, 3}, {4, 1}}; // input matrix 2

int row1 = mat1.length;

int col1 = mat1[0].length;

int col2 = mat2[0].length;

int[][] matSum = new int[row1][col2];

for(int i=0; i<row1; i++) {

for(int j=0; j<col2; j++) {

for(int k=0; k<col1; k++) {

matSum[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

// Printing the resultant matrix

System.out.println("Matrix multiplication result:");

for(int i=0; i<row1; i++) {

for(int j=0; j<col2; j++) {

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

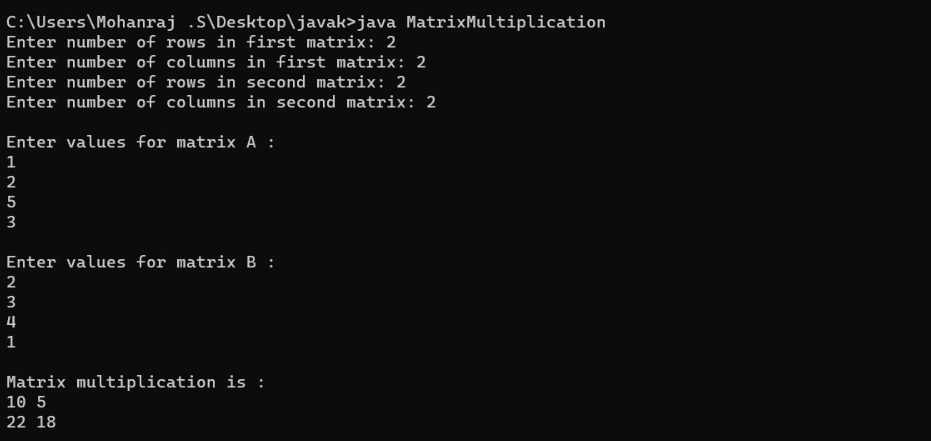
}

System.out.println();

}

}

}



14.public class MatrixAddition {

public static void main(String[] args) {

int[][] mat1 = {{1, 2}, {5, 3}}; // input matrix 1

int[][] mat2 = {{2, 3}, {4, 1}}; // input matrix 2

int row = mat1.length;

int col = mat1[0].length;

int[][] matSum = new int[row][col];

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

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

matSum[i][j] = mat1[i][j] + mat2[i][j];

}

}

// Printing the resultant matrix

System.out.println("Matrix addition result:");

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

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

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

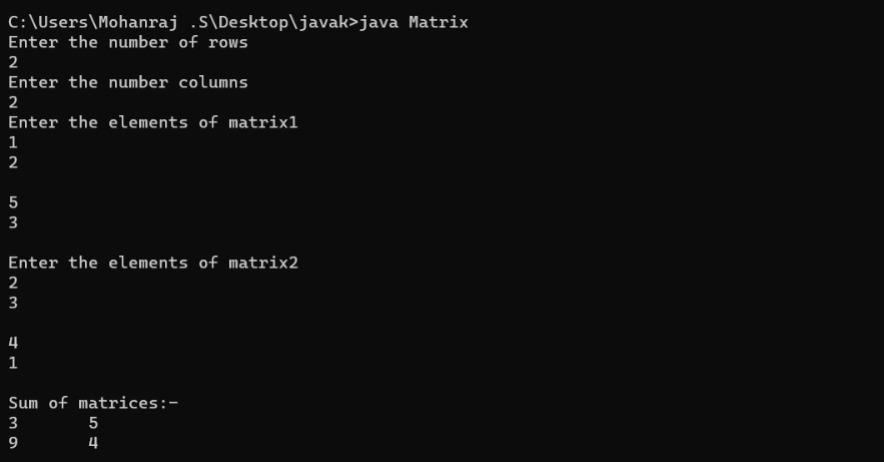
}

System.out.println();

}

}

}



15.

import java.util.Arrays;

import java.util.HashMap;

public class MeanMedianMode {

public static void main(String[] args) {

int[] arr = {16, 18, 27, 16, 23, 21, 19};

int n = arr.length;

// Mean

int sum = 0;

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

sum += arr[i];

}

double mean = (double)sum / n;

System.out.println("Mean = " + mean);

// Median

Arrays.sort(arr);

double median;

if(n % 2 == 0) {

median = (double)(arr[n/2] + arr[n/2 - 1]) / 2;

} else {

median = (double)arr[n/2];

}

System.out.println("Median = " + median);

// Mode

HashMap<Integer, Integer> freqMap = new HashMap<Integer, Integer>();

int maxFreq = 0;

int mode = 0;

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

int freq = freqMap.getOrDefault(arr[i], 0) + 1;

freqMap.put(arr[i], freq);

if(freq > maxFreq) {

maxFreq = freq;

mode = arr[i];

}

}

System.out.println("Mode = " + mode);

}

}

