**Assignment -2**

**Matrix Multiplication:**

public class MatrixMultiplication {

public static void main(String[] args) {

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

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

int[][] result = new int[2][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

result[i][j] = 0;

for (int k = 0; k < 2; k++) {

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

}

}

}

System.out.println("Resultant Matrix:");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

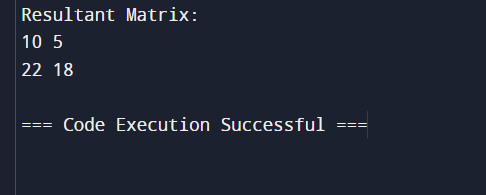
System.out.println();

}

}

}

Output:



Matrix Addition

public class MatrixAddition {

public static void main(String[] args) {

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

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

int[][] result = new int[2][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

}

System.out.println("Resultant Matrix:");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

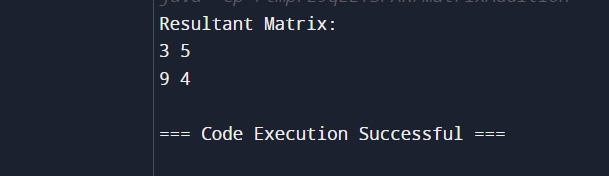
System.out.println();

}

}

}

Output:



Merge lists:

import java.util.ArrayList;

import java.util.Collections;

public class MergeArrays {

public static void main(String[] args) {

int[] arr1 = {1, 3, 4, 5};

int[] arr2 = {2, 4, 6, 8};

ArrayList<Integer> mergedList = new ArrayList<>();

for (int num : arr1) {

mergedList.add(num);

}

for (int num : arr2) {

mergedList.add(num);

}

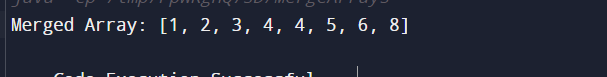
Collections.sort(mergedList);

System.out.println("Merged Array: " + mergedList);

}

}

Output:



Composite Number:

public class Main {

public static void main(String[] args) {

int[] numbers = {4, 7, 9, 10, 11, 12};

for (int num : numbers) {

if (isComposite(num)) {

System.out.println(num + " is a composite number.");

}

}

}

public static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i < num; i++) {

if (num % i == 0) {

return true;

}

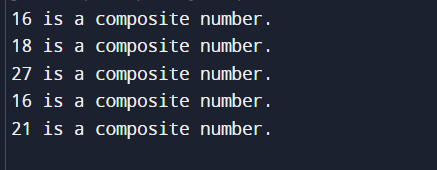
}

return false;

}

}

Output:



Right star pattern:

public class RightTrianglePattern {

public static void main(String[] args) {

int n = 5;

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

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

if (j < n - i - 1) {

System.out.print(" ");

} else {

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

}

}

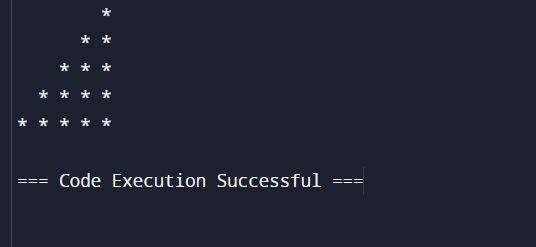
System.out.println();

}

}

}

Output:



Pascal Triangle:

public class PatternPrinting {

public static void main(String[] args) {

int rows = 5;

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

int number = 1;

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

System.out.print(" ");

}

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

System.out.print(" ");

if (j > 0) {

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

}

System.out.printf("%-3d", number);

}

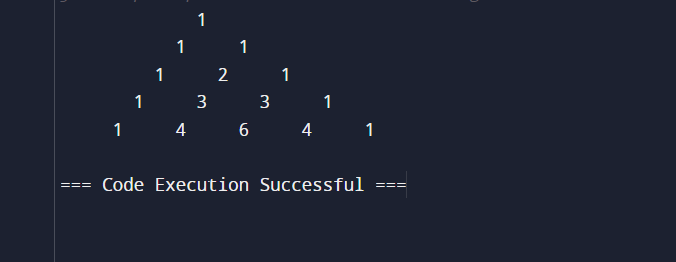
System.out.println();

}

}

}

Output:



Rectangular pattern

import java.util.Scanner;

public class RectanglePattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the symbol you want to use: ");

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

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

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

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

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

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

}

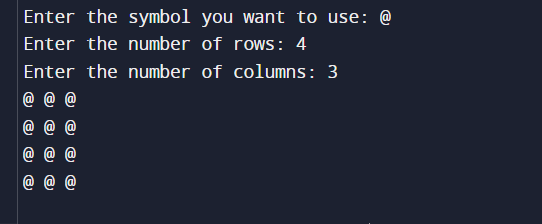
System.out.println();

}

}

}

Output:



Following pattern :

import java.util.Scanner;

public class PatternPrinting {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

System.out.print("Max Number of times to be printed: ");

int max = scanner.nextInt();

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);

}

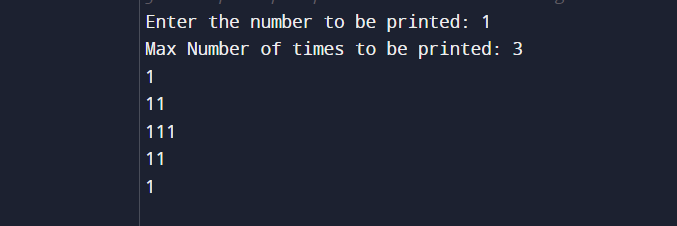
System.out.println();

}

}

}

Output:



Inverter full pyramid:

public class InvertedFullPyramid {

public static void main(String[] args) {

int rows = 5;

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

for (int space = 0; space < rows - i; ++space) {

System.out.print(" ");

}

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

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

}

for (int j = 0; j < i - 1; ++j) {

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

}

System.out.println();

}

}

}

Output:

