**1. Java Program to Find Even Sum of Fibonacci Series Till number N?**

**Sample Input: n = 4**

**Sample Output: 33**

**(N = 4, So here the fibonacci series will be produced from 0th term till 8th term:0, 1,**

**1, 2, 3, 5, 8, 13, 21**

**Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33)**

public class EvenSumFibonacci {

public static void main(String[] args) {

int n = 4;

int sum = findEvenIndexSum(n);

System.out.println("Sum of numbers at even indices: " + sum);

}

public static int findEvenIndexSum(int n) {

int a = 0, b = 1;

int sum = 0;

int index = 0;

while (a <= n) {

if (index % 2 == 0) {

sum += a;

}

int next = a + b;

a = b;

b = next;

index++;

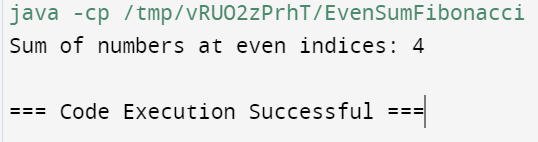
}

return sum;

}

}

output:



**2.Write a program to print the numbers from M to N by skipping K**

**numbers in between?**

**Sample Input:**

**M = 50**

**N = 100**

**K = 7**

**Sample Output:**

**50, 58, 66, 74, …..**

public class PrintNumbersSkipping {

public static void main(String[] args) {

int M = 50;

int N = 100;

int K = 7;

printNumbers(M, N, K);

}

public static void printNumbers(int M, int N, int K) {

for (int current = M; current <= N; current += (K + 1)) {

System.out.print(current);

if (current + (K + 1) <= N) {

System.out.print(", ");

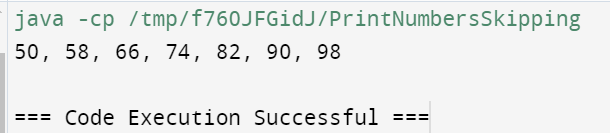
}

}

System.out.println(); }

}

output:



**3.Write a program for matrix addition?**

**Sample Input:**

**Mat1 = 1 2**

**5 3**

**Mat2 = 2 3**

**4 1**

**Sample Output:**

**Mat Sum = 3 5**

**9 4**

public class MatrixAddition {

public static void main(String[] args) {

int[][] mat1 = {

{1, 2},

{5, 3}

};

int[][] mat2 = {

{2, 3},

{4, 1}

};

int[][] matSum = addMatrices(mat1, mat2);

printMatrix(matSum);

}

public static int[][] addMatrices(int[][] mat1, int[][] mat2) {

int rows = mat1.length;

int cols = mat1[0].length;

int[][] result = new int[rows][cols];

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

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

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

}}

return result;

}

public static void printMatrix(int[][] matrix) {

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

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

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

}

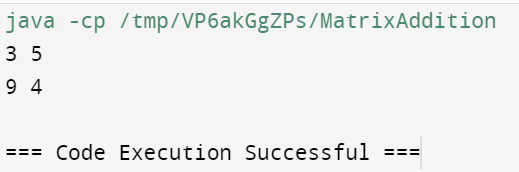
System.out.println();

}

}

}

output:



**4. Write a program for matrix multiplication?**

**Sample Input:**

**Mat1 = 1 2**

**5 3**

**Mat2 = 2 3**

**4 1**

**Sample Output:**

**Mat Sum = 10 5**

**22 18**

public class MatrixMultiplication {

public static void main(String[] args) {

int[][] mat1 = {

{1, 2},

{5, 3}

};

int[][] mat2 = {

{2, 3},

{4, 1}

};

int[][] matProduct = multiplyMatrices(mat1, mat2);

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

printMatrix(matProduct);

}

public static int[][] multiplyMatrices(int[][] mat1, int[][] mat2) {

int rowsMat1 = mat1.length;

int colsMat1 = mat1[0].length;

int rowsMat2 = mat2.length;

int colsMat2 = mat2[0].length;

if (colsMat1 != rowsMat2) {

throw new IllegalArgumentException("Number of columns in mat1 must be equal to number of rows in mat2.");

}

int[][] result = new int[rowsMat1][colsMat2];

for (int i = 0; i < rowsMat1; i++) {

for (int j = 0; j < colsMat2; j++) {

result[i][j] = 0;

for (int k = 0; k < colsMat1; k++) {

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

}

}

}

return result; }

public static void printMatrix(int[][] matrix) {

for (int[] row : matrix) {

for (int value : row) {

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

}

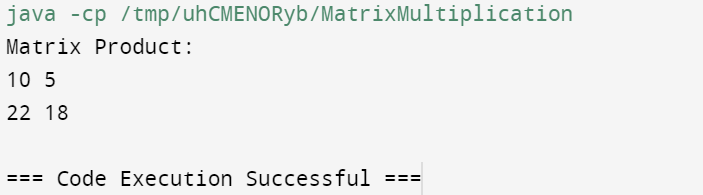
System.out.println();

}

}

}

output:



**5.Write a program to print the following pattern**

**Sample Input:**

**Enter the number to be printed: 1**

**Max Number of time printed: 3**

**1**

**11**

**111**

**11**

**1**

import java.util.Scanner;

public class PatternPrinter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

System.out.print("Max number of times printed: ");

int maxCount = scanner.nextInt();

printPattern(number, maxCount);

scanner.close();

}

public static void printPattern(char number, int maxCount) {

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

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

System.out.print(number);

}

System.out.println();

}

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

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

System.out.print(number);

}

System.out.println();

}

}

}

output:

