|  |  |
| --- | --- |
| **Ex. No. 2** | **ARRAYS AND ARRAY OF OBJECTS** |
| **Date of Exercise** | 20-07-2023 |

1.**Aim:**

To write a java program to implement the following procedure to generate prime numbers from 1 to 100 into a program. This procedure is called sieve of Eratosthenes.

**Procedure:**

Step1:start the program

Step2: Create a boolean array "prime[100]" and initialize all entries as true.

Step3: Mark all the multiples of each prime number as false

Step4: Copy Numbers to Two Arrays (Ascending and Descending)

Step5:stop the program

**Program:**

public class SieveOfEratosthenes {

public static void main(String[] args) {

int limit = 100;

boolean[] primes = new boolean[limit + 1];

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

primes[i] = true;

System.out.println(primes[i]);

}

for (int num = 2; num \* num <= limit; num++) {

if (primes[num]) {

for (int multiple = num \* num; multiple <= limit; multiple += num) {

primes[multiple] = false;

}

}

}

System.out.println("Prime numbers from 1 to " + limit + ":");

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

if (primes[i]) {

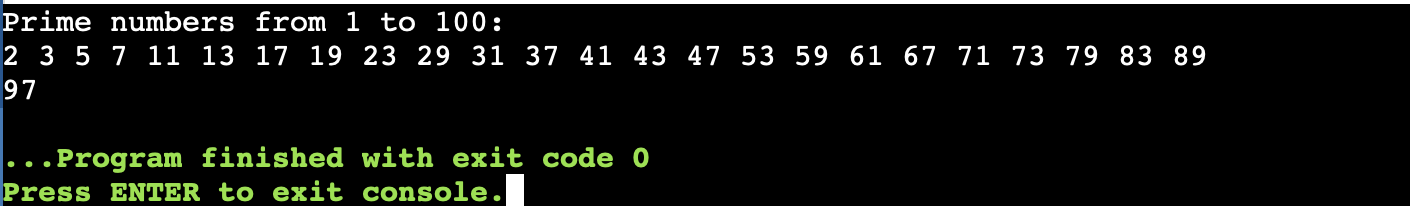
System.out.print(i + " ");

}

}

}

}

**Output:**

**Result:**

The above program has been successfully executed and verified.

2.**Aim:**

To write a java program to copy the given N numbers of one array into another 2 arrays in such a way that one array must contain the numbers in ascending order and the other must contain in the descending order.

**Procedure:**

Step1:start the program

Step2: Initialize the source array with N numbers.

Step3: Create two new arrays with the same size

Step4: Iterate through the sorted array and fill the descending array in reverse order.

Step5:stop the program

**Program:**

import java.util.Arrays;

import java.util.Scanner;

public class hi {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] orgarr = {5, 7, 4};

int[] ascendingArray = new int[3];

int[] decendingArray = new int[3];

System.out.println("Enter the elements of the source array:");

for (int i = 0; i < 3; i++) {

ascendingArray[i] = orgarr[i];

}

for (int i = 0; i < 3; i++) {

decendingArray[i] = orgarr[i];

}

Arrays.sort(ascendingArray);

Arrays.sort(decendingArray);

System.out.println("Sorted array in ascending order:");

for (int i = 0; i < 3; i++) {

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

}

System.out.println();

System.out.println("Sorted array in descending order:");

for (int i = 2; i >= 0; i--) {

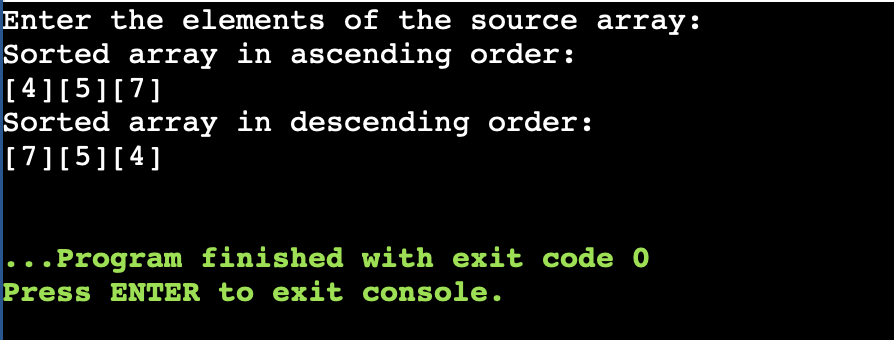
System.out.print("[" + decendingArray[i] + "]");

}

System.out.println();

}

}

**Output:**

**Result:**

The above program has been successfully executed and verified.

3.**Aim:**

To write a java program to obtain transpose of a 3 x 3 matrix. The transpose of a matrix is obtained by exchanging the elements of each row with the elements of the corresponding column.

**Procedure:**

Step1:start the program

Step2: Create a 3x3 matrix and initialize it with values.

Step3: Create a new 3x3 matrix to store the transpose.

Step4: The transpose matrix now contains the transpose of the original matrix.

Step5:stop the program

**Program:**

import java.util.Scanner;

public class hi {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[][] matrix = new int[3][3];

System.out.println("Enter the 3x3 matrix (9 integers in total):");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

matrix[i][j] = scanner.nextInt();

}

}

int[][] transpose = findTranspose(matrix);

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

displayMatrix(matrix);

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

displayMatrix(transpose);

}

public static int[][] findTranspose(int[][] matrix) {

int[][] transpose = new int[3][3];

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

transpose[i][j] = matrix[j][i];

}

}

return transpose;

}

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

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

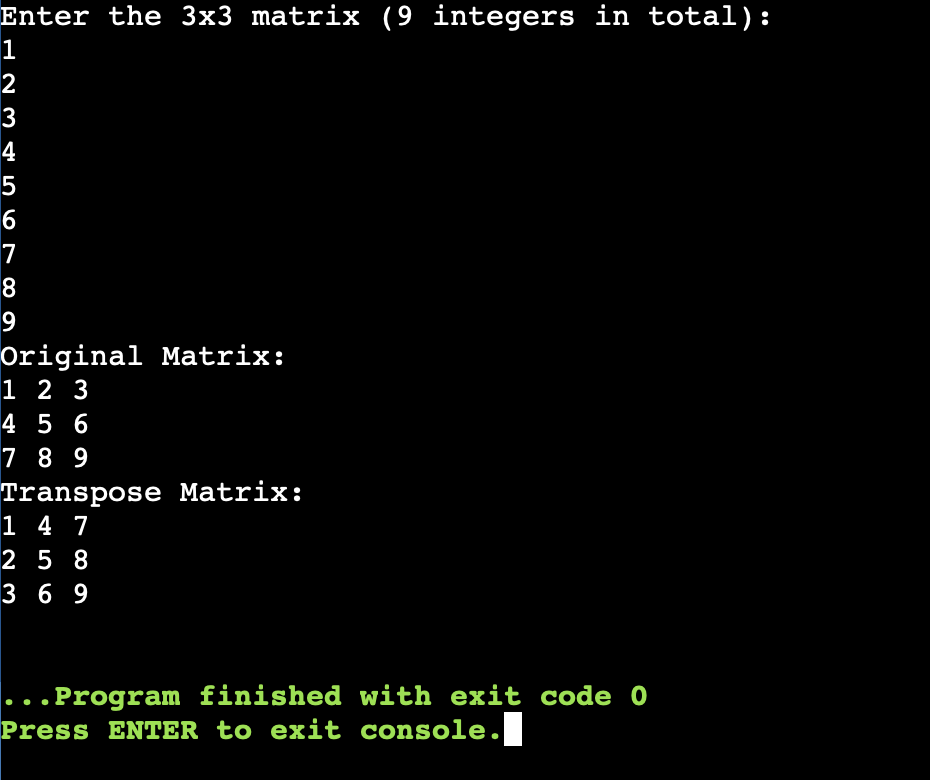
System.out.println();

}

}

}

**Output:**

****

**Result:**

The above program has been successfully executed and verified.

4.**Aim:**

To write a Java program to find the common elements between two arrays.

**Procedure:**

Step1:start the program

Step2: Create two arrays of N elements each.

Step3: Sort both arrays using Arrays.sort() or any sorting algorithm.

Step4: Compare the elements at the current positions of both pointers.

Step5:stop the program

**Program:**

public class CommonElementsArrays {

public static void main(String[] args) {

int[] array1 = {1, 2, 3, 4, 5};

int[] array2 = {3, 4, 5, 6, 7};

System.out.println("Common elements between the two arrays:");

findCommonElements(array1, array2);

}

public static void findCommonElements(int[] array1, int[] array2) {

for (int i = 0; i < array1.length; i++) {

for (int j = 0; j < array2.length; j++) {

if (array1[i] == array2[j]) {

System.out.print(array1[i] + " ");

break;

}

}

}

}

}

**Output:**

**A black background with white text

Description automatically generated**

**Result:**

The above program has been successfully executed and verified.

5.**Aim:**

To write a program in Java to count the occurrence of a given character in an array

**Procedure:**

Step1:start the program

Step2: Create an array of characters with N elements and initialize it.

Step3: Input the character to be counted.

Step4: Use a loop to iterate through the array elements.

Step5:stop the program

**Program:**

import java.util.Scanner;

public class hi {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the array of characters (without spaces): ");

String input = scanner.nextLine();

char[] charArray = input.toCharArray();

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

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

int count = countOccurrences(charArray, targetCharacter);

System.out.println("The character '" + targetCharacter + "' occurs " + count + " time(s) in the array."); }

public static int countOccurrences(char[] array, char target) {

int count = 0;

for (char c : array) {

if (c == target) {

count++; }

}

return count;

}

}

**Output:**

**A screen shot of a computer code

Description automatically generated**

**Result:**

The above program has been successfully executed and verified.