**Nishant Patil**

**PG-DAC**

**Java**

**Assignment 3**

***1. Find the Largest and Smallest Element***

***○ Given an array, find the smallest and largest elements in it.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

int largest = arr[0], smallest = arr[0];

for (int i = 1; i < arr.length; i++) {

if (arr[i] > largest) largest = arr[i];

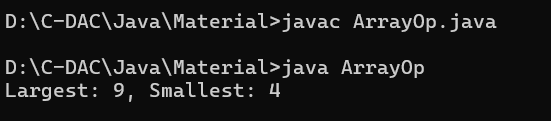
if (arr[i] < smallest) smallest = arr[i];

}

System.out.println("Largest: " + largest + ", Smallest: " + smallest);

}

}



***2. Reverse an Array***

***○ Reverse the given array in place.***

import java.util.Arrays;

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

int left = 0, right = arr.length - 1;

while (left < right) {

int temp = arr[left];

arr[left] = arr[right];

arr[right] = temp;

left++;

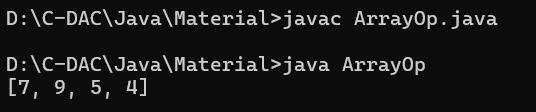
right--;

}

System.out.println(Arrays.toString(arr));

}

}



***3. Find the Second Largest Element***

***○ Find the second-largest element in the given array.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

int largest = Integer.MIN\_VALUE, secondLargest = Integer.MIN\_VALUE;

for (int num : arr) {

if (num > largest) {

secondLargest = largest;

largest = num;

} else if (num > secondLargest && num != largest) {

secondLargest = num;

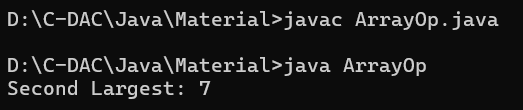
}

}

System.out.println("Second Largest: " + secondLargest);

}

}



***4. Count Even and Odd Numbers***

***○ Count the number of even and odd numbers in an array.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

int even = 0, odd = 0;

for (int num : arr) {

if (num % 2 == 0) even++;

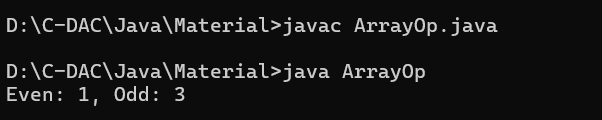
else odd++;

}

System.out.println("Even: " + even + ", Odd: " + odd);

}

}



***5. Find Sum and Average***

***○ Compute the sum and average of all elements in the array.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

int sum = 0;

for (int num : arr) {

sum += num;

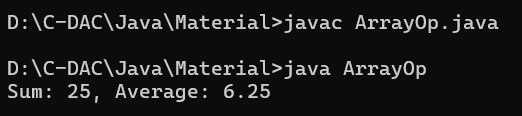
}

double average = sum / (double) arr.length;

System.out.println("Sum: " + sum + ", Average: " + average);

}

}



***6. Remove Duplicates from a Sorted Array***

***○ Remove duplicate elements from a sorted array without using extra space.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 5, 7, 9,};

if (arr.length == 0) {

System.out.println("Array is empty.");

return;

}

int j = 0;

for (int i = 1; i < arr.length; i++) {

if (arr[i] != arr[j]) {

j++;

arr[j] = arr[i];

}

}

System.out.print("Array with unique elements: ");

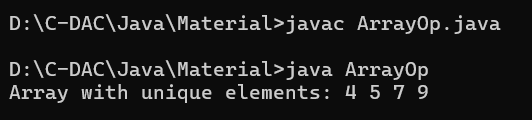
for (int i = 0; i <= j; i++) {

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

}

System.out.println();

}

}

}***7. Rotate an Array***

***○ Rotate the array to the right by k positions.***

import java.util.Scanner;

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 4, 5, 9, 7 };

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rotations: ");

int k = sc.nextInt();

k = k % arr.length;

reverseArr(arr, 0, arr.length - 1);

reverseArr(arr, 0, k - 1);

reverseArr(arr, k, arr.length - 1);

System.out.print("Array after " + k + " rotations: ");

printArr(arr);

}

private static void reverseArr(int[] arr, int start, int end) {

while (start < end) {

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

private static void printArr(int[] arr) {

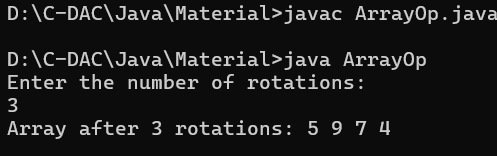
for (int i = 0; i < arr.length; i++) {

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

}

System.out.println();

}

}

***8. Merge Two Sorted Arrays***

***○ Merge two sorted arrays into a single sorted array without using extra space.***

import java.util.Arrays;

public class ArrayOp {

public static void main(String args[]) {

int[] arr1 = new int[] { 4, 5, 7, 9 };

int[] arr2 = new int[] { 1, 3, 6,};

int[] result = new int[arr1.length + arr2.length];

int i = 0, j = 0, k = 0;

while (i < arr1.length && j < arr2.length) {

if (arr1[i] < arr2[j]) {

result[k++] = arr1[i++];

}

else result[k++] = arr2[j++];

}

while (i < arr1.length){

result[k++] = arr1[i++];

}

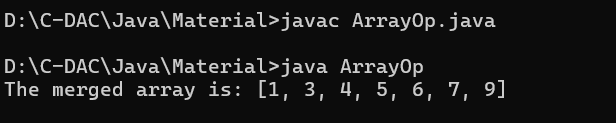
while (j < arr2.length){

result[k++] = arr2[j++];

}

System.out.println("The merged array is: " + Arrays.toString(result));

}

}

***9. Find Missing Number in an Array***

***○ Given an array of size n-1 containing numbers from 1 to n, find the missing number.***

public class ArrayOp {

public static void main(String args[]) {

int[] arr = new int[] { 1, 2, 3, 4, 5, 7, 9 };

int n = arr.length + 1;

int totalSum = n \* (n + 1) / 2;

int arrSum = 0;

for (int num : arr) {

arrSum += num;

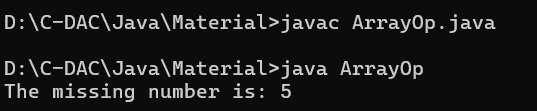
}

int miss = totalSum - arrSum;

System.out.println("The missing number is: " + miss);

}

}



***10. Find Intersection and Union of Two Arrays***

***○ Find the intersection and union of two unsorted arrays.***

public class ArrayOp {

public static void interun(int[] arr1, int[] arr2) {

System.out.print("Union: ");

for (int i = 0; i < arr1.length; i++) {

System.out.print(arr1[i] + " ");

}

for (int i = 0; i < arr2.length; i++) {

boolean alreadyExists = false;

for (int j = 0; j < arr1.length; j++) {

if (arr2[i] == arr1[j]) {

alreadyExists = true;

break;

}

}

if (!alreadyExists) {

System.out.print(arr2[i] + " ");

}

}

System.out.println();

System.out.print("Intersection: ");

for (int i = 0; i < arr1.length; i++) {

for (int j = 0; j < arr2.length; j++) {

if (arr1[i] == arr2[j]) {

System.out.print(arr1[i] + " ");

break;

}

}

}

System.out.println();

}

public static void main(String[] args) {

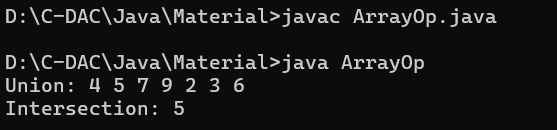
int[] arr1 = {4, 5, 7, 9};

int[] arr2 = {2, 3, 5, 6};

interun(arr1, arr2);

}

}



***11. Find a Subarray with Given Sum***

***○ Given an array of integers, find the subarray that sums to a given value S.***

public class ArrayOp{

public static void subarray (int[] arr, int sum) {

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

int currentSum = arr[i];

if (currentSum == sum) {

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

return;

}

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

currentSum += arr[j];

if (currentSum == sum) {

System.out.print("Subarray: [");

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

System.out.print(arr[k]);

if (k < j) {

System.out.print(", "); }

}

System.out.println("]");

return;

}

}

}

System.out.println("No subarray found with the given sum");

}

public static void main(String[] args) {

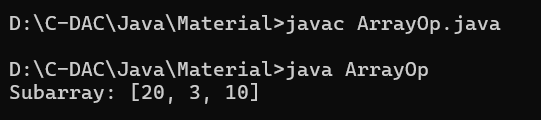
int[] arr = {1, 4, 20, 3, 10, 5};

int sum = 33;

subarray (arr, sum);

}

}



***12. Write a program to accept 20 integer numbers in a single Dimensional Array. Find and***

***Display the following:***

***○ Number of even numbers.***

***○ Number of odd numbers.***

***○ Number of multiples of 3***

public class ArrayOp {

public static void countNum(int[] arr) {

int even = 0, odd = 0, multi3 = 0;

for (int num : arr) {

if (num % 2 == 0) {

even++;

} else {

odd++;

}

if (num % 3 == 0) {

multi3++;

}

}

System.out.println("Even Numbers: " + even);

System.out.println("Odd Numbers: " + odd);

System.out.println("Multiples of 3: " + multi3);

}

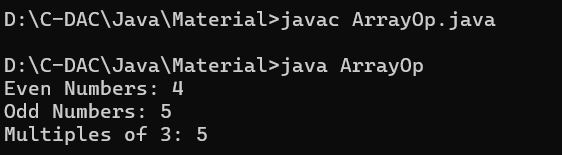
public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5, 6, 9, 12, 15};

countNum(arr);

}

}



***13. Write a program to accept the marks in Physics, Chemistry and Maths secured by 20 class students in a single Dimensional Array. Find and display the following:***

***○ Number of students securing 75% and above in aggregate.***

***○ Number of students securing 40% and below in aggregate.***

import java.util.Scanner;

public class ArrayOp {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int numStudents = 20;

int[] marks = new int[numStudents \* 3];

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

System.out.println("Enter marks for Student " + (i + 1) + ":");

System.out.print("Physics: ");

marks[i \* 3] = sc.nextInt();

System.out.print("Chemistry: ");

marks[i \* 3 + 1] = sc.nextInt();

System.out.print("Maths: ");

marks[i \* 3 + 2] = sc.nextInt();

}

int countAbove75 = 0;

int countBelow40 = 0;

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

int physics = marks[i \* 3];

int chemistry = marks[i \* 3 + 1];

int maths = marks[i \* 3 + 2];

int totalMarks = physics + chemistry + maths;

double percentage = (double) totalMarks / 3;

if (percentage >= 75) {

countAbove75++;

} else if (percentage <= 40) {

countBelow40++;

}

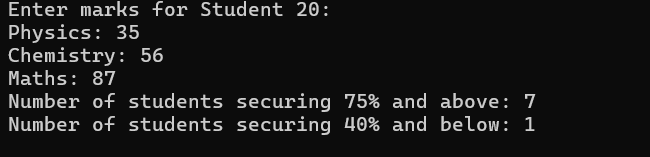
}

System.out.println("Number of students securing 75% and above: " + countAbove75);

System.out.println("Number of students securing 40% and below: " + countBelow40);

sc.close();

}

}

***14. Write a program in Java to accept 20 numbers in a single dimensional array arr[20]. Transfer and store all the even numbers in an array even[ ] and all the odd numbers in another array odd[ ]. Finally, print the elements of the even & the odd array.***

import java.util.Scanner;

public class ArrayOp {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = new int[20];

int[] even = new int[20];

int[] odd = new int[20];

int evenIndex = 0, oddIndex = 0;

System.out.println("Enter 20 numbers:");

for (int i = 0; i < 20; i++) {

arr[i] = sc.nextInt();

}

for (int i = 0; i < 20; i++) {

if (arr[i] % 2 == 0) {

even[evenIndex++] = arr[i];

} else {

odd[oddIndex++] = arr[i];

}

}

System.out.print("Even Numbers: ");

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

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

}

System.out.println();

System.out.print("Odd Numbers: ");

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

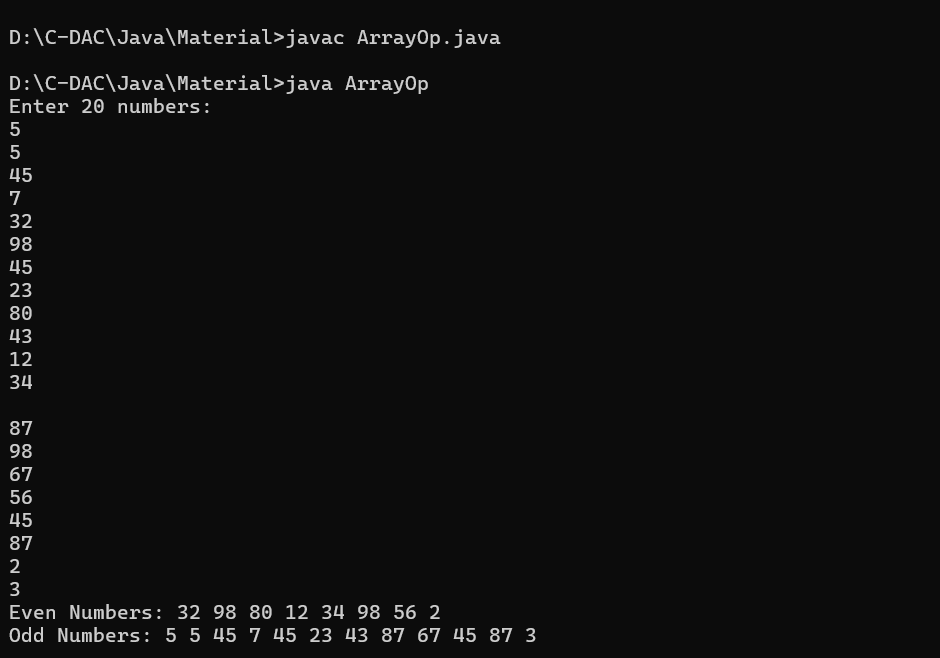
System.out.print(odd[i] + " ");

}

System.out.println();

}

}

***15. Write a Java program to print all sub-arrays with 0 sum present in a given array of integers.***

***Example:***

***Input :***

***nums1 = { 1, 3, -7, 3, 2, 3, 1, -3, -2, -2 }***

***nums2 = { 1, 2, -3, 4, 5, 6 }***

***nums3= { 1, 2, -2, 3, 4, 5, 6 }***

***Output:***

***Sub-arrays with 0 sum : [1, 3, -7, 3]***

***Sub-arrays with 0 sum : [3, -7, 3, 2, 3, 1, -3, -2]***

***Sub-arrays with 0 sum : [1, 2, -3]***

***Sub-arrays with 0 sum : [2, -2]***

public class ArrayOp {

public static void zeroSum(int[] nums) {

int n = nums.length;

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

int sum = 0;

for (int end = start; end < n; end++) {

sum += nums[end];

if (sum == 0) {

System.out.print("Sub-array with 0 sum: [");

for (int i = start; i <= end; i++) {

System.out.print(nums[i]);

if (i < end) System.out.print(", ");

}

System.out.println("]");

}

}

}

}

public static void main(String[] args) {

int[] nums1 = {1, 3, -7, 3, 2, 3, 1, -3, -2, -2};

int[] nums2 = {1, 2, -3, 4, 5, 6};

int[] nums3 = {1, 2, -2, 3, 4, 5, 6};

System.out.println("Subarrays with 0 sum in nums1:");

zeroSum(nums1);

System.out.println("\nSubarrays with 0 sum in nums2:");

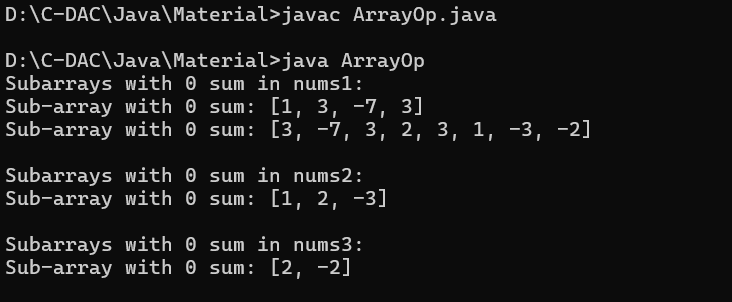
zeroSum(nums2);

System.out.println("\nSubarrays with 0 sum in nums3:");

zeroSum(nums3);

}

}

***16. Given two sorted arrays A and B of size p and q, write a Java program to merge elements of***

***A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B***

***with remaining elements.***

***Example:***

***Input :***

***int[] A = { 1, 5, 6, 7, 8, 10 }***

***int[] B = { 2, 4, 9 }***

***Output:***

***Sorted Arrays:***

***A: [1, 2, 4, 5, 6, 7]***

***B: [8, 9, 10]***

import java.util.Arrays;

public class ArrayOp {

public static void main(String[] args) {

int[] A = { 1, 5, 6, 7, 8, 10 };

int[] B = { 2, 4, 9 };

mergeAndSort(A, B);

// Displaying the result

System.out.println("Sorted Arrays:");

System.out.println("A: " + Arrays.toString(A));

System.out.println("B: " + Arrays.toString(B));

}

public static void mergeAndSort(int[] A, int[] B) {

// First, combine both arrays A and B into a single array

int totalLength = A.length + B.length;

int[] combined = new int[totalLength];

System.arraycopy(A, 0, combined, 0, A.length); // Copy elements of A into combined

System.arraycopy(B, 0, combined, A.length, B.length); // Copy elements of B into combined

// Sort the combined array

Arrays.sort(combined);

// Now, split the combined sorted array into A and B

int p = A.length;

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

A[i] = combined[i]; // Fill A with the first p smallest elements

}

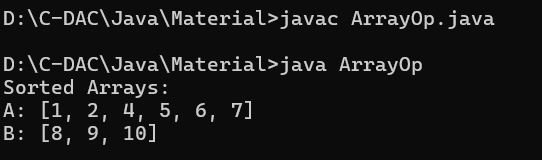
for (int i = 0; i < B.length; i++) {

B[i] = combined[p + i]; // Fill B with the remaining elements

}

}

}

******

***17. Write a Java program to find the maximum product of two integers in a given array of***

***integers.***

***Example:***

***Input :***

***nums = { 2, 3, 5, 7, -7, 5, 8, -5 }***

***Output:***

***Pair is (7, 8), Maximum Product: 56***

public class ArrayOp {

public static void main(String[] args) {

int[] nums = {2, 3, 5, 7, -7, 5, 8, -5};

maxPro(nums);

}

public static void maxPro(int[] nums) {

int maxProduct = Integer.MIN\_VALUE;

int num1 = 0, num2 = 0;

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

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

int product = nums[i] \* nums[j];

if (product > maxProduct) {

maxProduct = product;

num1 = nums[i];

num2 = nums[j];

}

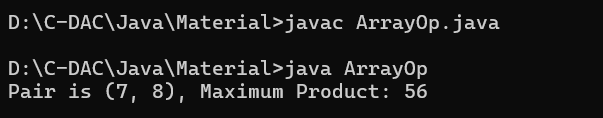
}

}

System.out.println("Pair is (" + num1 + ", " + num2 + "), Maximum Product: " + maxProduct);

}

}

******

***18. Print a Matrix***

***○ Given an m x n matrix, print all its elements row-wise.***

public class ArrayOp{

public static void main(String[] args) {

int[] a={1,2,3};

int[] b={4,5,6};

int arr[][]=new int[2][];

arr[0]=a;

arr[1]=b;

for(int ar[]: arr){

for(int x:ar){

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

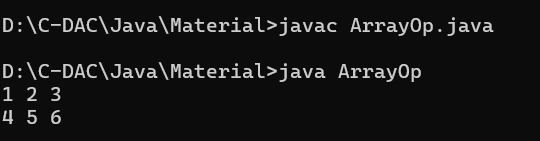
}

System.out.println();

}

}

}



***19. Transpose of a Matrix***

***○ Given a matrix, return its transpose (swap rows and columns).***

public class ArrayOp {

public static void main(String[] args) {

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

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

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

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

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

}

System.out.println();

}

}

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

int rows = matrix.length;

int cols = matrix[0].length;

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

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

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

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

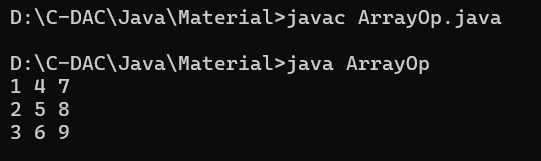
}

}

return transpose;

}

}

******

***20. Sum of Two Matrices***

***○ Given two matrices of the same size, compute their sum.***

public class ArrayOp {

public static void main(String args[]){

int[][] a={{1,2},{1,5}};

int[][] b={{5,6},{1,3}};

int [][]sum=new int[2][];

sum[0]=new int[2];

sum[1]=new int[2];

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

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

sum[i][j] = a[i][j]+b[i][j];

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

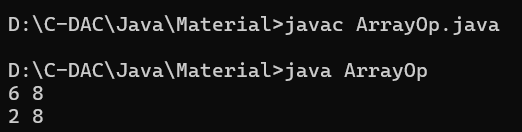
}

System.out.println();

}

}

}

******

***21. Row-wise and Column-wise Sum***

***○ Find the sum of each row and each column of a given matrix.***

public class ArrayOp {

public static void main(String[] args) {

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

System.out.println("The row sum: ");

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

int rowsum=0;

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

rowsum=rowsum+a[i][j] ;

}

System.out.println(rowsum);

}

System.out.println("The column sum: ");

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

int colsum=0;

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

colsum=colsum+a[i][j] ;

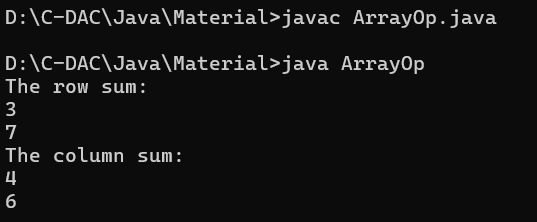
}

System.out.println(colsum);

}

}

}

******

**22. Find the Maximum Element in a Matrix**

**○ Find the largest element in a given matrix.**

public class ArrayOp {

public static void main(String[] args) {

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int maxElement = maxEl(matrix);

System.out.println("Maximum element in the matrix: " + maxElement);

}

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

int max = matrix[0][0];

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

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

if (matrix[i][j] > max) {

max = matrix[i][j];

}

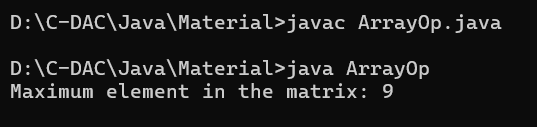
}

}

return max;

}

}

****

***23. Matrix Multiplication***

***○ Multiply two matrices and return the resultant matrix.***

public class ArrayOp {

public static void main(String[] args) {

int[][] matrix1 = {

{1, 2},

{3, 4}

};

int[][] matrix2 = {

{5, 6},

{7, 8}

};

int[][] product = multiply(matrix1, matrix2);

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

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

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

}

System.out.println();

}

}

public static int[][] multiply(int[][] matrix1, int[][] matrix2) {

int rows1 = matrix1.length;

int cols1 = matrix1[0].length;

int rows2 = matrix2.length;

int cols2 = matrix2[0].length;

int[][] result = new int[rows1][cols2];

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

result[i][j] = 0;

for (int k = 0; k < cols1; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

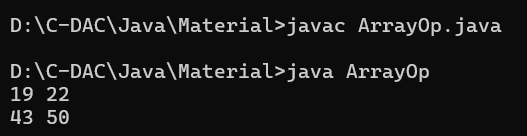
}

}

return result;

}

}

******

***24. Rotate a Matrix by 90 Degrees***

***○ Rotate a given N x N matrix by 90 degrees clockwise.***

import java.util.Scanner;

public class ArrayOp{

public static void main (String[] args){

Scanner scr = new Scanner(System.in);

int[][] a = new int[3][];

a[0] = new int[3];

a[1] = new int[3];

a[2] = new int[3];

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

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

System.out.println("Enter an element : ");

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

}

}

System.out.println("\nYour array is: ");

for(int[] b : a){

for(int c : b){

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

}

System.out.println();

}

{

int[] l = new int[3];

l=a[0];

a[0]=a[2];

a[2]=l;

}

{

System.out.println("\nThe rotated array :");

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

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

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

}

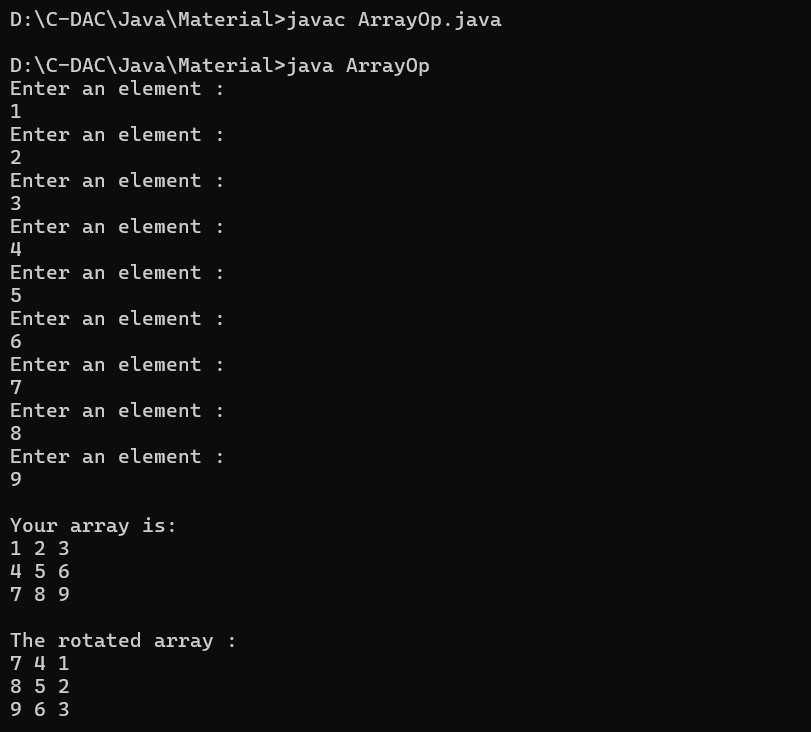
System.out.println();

}

}

}

}

******

***25. Find the Diagonal Sum***

***○ Compute the sum of both diagonals in a square matrix.***

public class ArrayOp {

public static void main(String[] args) {

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int sum = diagonalSum(matrix);

System.out.println("Diagonal Sum: " + sum);

}

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

int sum = 0;

int n = matrix.length;

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

sum += matrix[i][i];

sum += matrix[i][n - i - 1];

}

if (n % 2 != 0) {

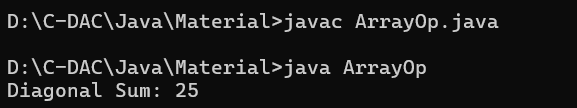
sum -= matrix[n / 2][n / 2];

}

return sum;

}

}

******