**EX NO:4 ARRAYS**

**DATE: 23/08/2023**

**AIM**

To explore the Arrays in Java and performs basic operations like sorting.

**ALGORITHM:**

**STEP 1:** To displaying the sorted array format. Include O(n^2) and O(n) complexity sorting algorithms, and print the comparisons for each.

1. **int[] arr** = new int[n] - To initialize an integer array to store marks for each subject.
2. **bubbleSort(int[] arr)** - Sorts an array in ascending order using the Bubble Sort algorithm and returns the number of comparisons.
3. **selectionSort(int[] arr)** - Sorts an array in ascending order using the Selection Sort algorithm and returns the number of comparisons.
4. **printArray(int[] arr)** - Prints the elements of an array in a formatted manner.

**STEP 2:** To create a Java program with static methods to read and sort n random integers, returning comparisons. Also, generate and sort a character array from the random integers.

1. **RandomArray(int[] arr)**- Generates an array of random integers between 0 and 25..
2. **Random rand = new Random()** - To create a random number generator.
3. **rand.nextInt(int)** - To generate a random integer within the specified range.
4. **CharArray(int[] intArray)** - Converts an integer array to a character array, mapping integers to corresponding lowercase English letters.
5. **OrderNSort(char[] arr)** - Sorts a character array in ascending order using the Selection Sort algorithm and returns the number of comparisons.
6. **OrderNSort(int[] arr)** - Sorts an integer array in ascending order using the Selection Sort algorithm and returns the number of comparisons.

**STEP 3:** To write Java program to read n random integers into arrays A and B, both of size n. Use a method to search and print the occurrences of each element from B in A.

1. **Random rand = new Random()** - To create a random number generator.
2. **rand.nextInt(int)** - To generate a random integer within the specified range.
3. **countOccurrences(int[] arr, int target)** - Counts the occurrences of a specified target integer in an array and returns the count.

**STEP 4:** To calculates column-wise sums using a method, and sorts the arrays based on the column sums.

**1. readMatrix(Scanner scanner, int[][] matrix)** - Reads input elements for a matrix from the user using a scanner.

**2. addMatrices(int[][] matrix1, int[][] matrix2)** - Adds two matrices and returns the result.

**3. sortMatrixByColumnSum(int[][] matrix)** - Sorts the matrix by the sum of each column in ascending order.

**4. printMatrix(int[][] matrix)** - Prints the elements of a matrix in a formatted manner.

1. **Write a program to read n integer in a 1D array and print the sorted array in the following format. Use static methods and find the number of comparisons for the sorting algorithm whose worst-case complexity is O(n^2) and O(n) Print the array with array index position.**

**Program.**

import java.util.Scanner;

public class SortedArray3704 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] arr = new int[n];

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

System.out.print("Enter element " + (i + 1) + ": ");

arr[i] = scanner.nextInt();

}

int comparisons = bubbleSort(arr);

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

printSortedArrayWithIndex(arr);

System.out.println("Number of comparisons: " + comparisons);

}

public static int bubbleSort(int[] arr) {

int n = arr.length;

int comparisons = 0;

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

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

comparisons++;

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

} } }

return comparisons; }

public static void printSortedArrayWithIndex(int[] arr) {

System.out.println("-----------------------------------------------");

System.out.print("|");

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

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

}

System.out.println("\n---------------------------");

System.out.print("|");

for (int num : arr) {

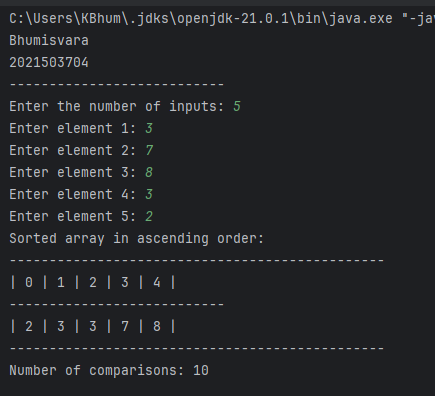
System.out.print(" " + num + " |");

}

System.out.println("\n-----------------------------------------------");

}

**OUTPUT.**



**2. Sort random integer/character Write a program to read n random integer in a 1D array.**

**A. Apply method to sort the generated array content and return the number of comparisons done.**

**Program.**

import java.util.Random;

import java.util.Scanner;

public class Random3704 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] intArray = new int[n];

intArray = randomArray(intArray);

System.out.println("Random Integer Array:");

printIntArray(intArray);

int intComparisons = bubbleSort(intArray);

System.out.println("Sorted Integer Array:");

printIntArray(intArray);

System.out.println("Number of comparisons: " + intComparisons);

}

public static int[] randomArray(int[] arr) {

Random random = new Random();

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

arr[i] = random.nextInt(100); //

} return arr;

}

public static int bubbleSort(int[] arr) {

int n = arr.length;

int comparisons = 0;

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

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

comparisons++;

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

} } } return comparisons;

}

public static void printIntArray(int[] arr) {

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

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

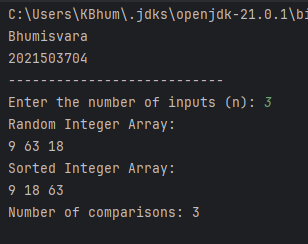
}

System.out.println();

}

}

**OUTPUT.**



**B. Apply another method to generate character array using the random integer and sort the array.**

**Program.**

import java.util.Random;

import java.util.Scanner;

public class RandomChar3704 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] intArray = new int[n];

intArray = randomArray(intArray);

System.out.println("Random Integer Array:");

printIntArray(intArray);

char[] charArray = charArray(intArray);

System.out.println("Generated Character Array:");

printCharArray(charArray);

bubbleSort(charArray);

System.out.println("Sorted Character Array:");

printCharArray(charArray);

}

public static int[] randomArray(int[] arr) {

Random random = new Random();

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

arr[i] = random.nextInt(26); // Generate random integer between 0 and 25

} return arr; }

public static char[] charArray(int[] intArray) {

char[] charArray = new char[intArray.length];

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

charArray[i] = (char) (intArray[i] + 'a'); // Convert integer to character 'a' to 'z'

}

return charArray;

}

public static void bubbleSort(char[] arr) {

int n = arr.length;

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

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

if (arr[j] > arr[j + 1]) {

char temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

} } } }

public static void printIntArray(int[] arr) {

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

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

}

System.out.println();

}

public static void printCharArray(char[] arr) {

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

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

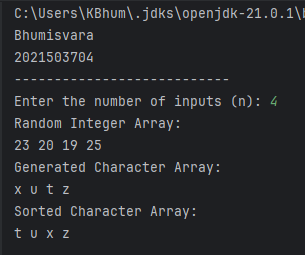
}

System.out.println();

}

}

**OUTPUT.**



1. **Write a program to read n random integer in a 1D array of A and B of size n. Apply method to search the occurrence of element in B and print the number of B element occurrence in A .**

**Source code:**

import java.util.Random;

import java.util.Scanner;

public Main3704 {

public static void main(String[] args)

Scanner scanner = new Scanner(System.in);

Random random = new Random();

int n = random.nextInt(10) + 1;

int[] A = new int[n];

int[] B = new int[n];

System.out.println("Randomly generated value of n: " + n);

System.out.println("Random integers for array A:");

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

A[i] = random.nextInt(100);

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

}

System.out.println();

System.out.println("Random integers for array B:");

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

B[i] = random.nextInt(100);

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

}

System.out.println();

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

// Count the occurrences of each element of B in array A

int count = countOccurrences(A, B[i]);

System.out.println("Number of occurrences of " + B[i] + " in array A: " + count);

}

}

public static int countOccurrences(int[] arr, int target) {

int count = 0;

for (int num : arr) {

if (num == target) {

count++;

}

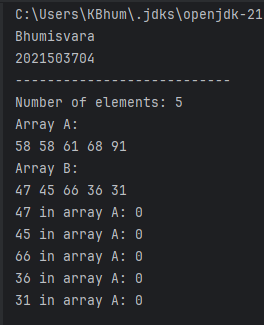
}

return count;

}

}

**OUTPUT**



1. **Write a program to read two 2D array. Apply method to perform column major sum and sort the array based on the sum of columns.**

**Source Code:**

import java.time.LocalDate;

import java.time.LocalTime;

import java.util.Scanner;

public Main3704 {

static void arraySum(int[][] a, int[][] s, int rows, int cols) {

// Calculate the sum of corresponding elements in two arrays

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

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

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

}

}

System.out.println("Sum of two arrays is:");

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

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

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

}

System.out.println();

}

}

static void colSumMax(int[][] x, int[][] y, int rows, int cols) {

// Calculate the sum of arrays and sort based on column sum

arraySum(x, y, rows, cols);

int[] colsum = new int[cols];

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

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

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

}

}

int[] csindex = new int[cols];

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

csindex[j] = j;

}

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

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

if (colsum[j] > colsum[j + 1]) {

int temp = colsum[j];

colsum[j] = colsum[j + 1];

colsum[j + 1] = temp;

int temp1 = csindex[j];

csindex[j] = csindex[j + 1];

csindex[j + 1] = temp1;

}

}

}

System.out.println("The array after sorting based on column sum is:");

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

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

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

}

System.out.println();

}

}

public static void main(String[] args) {

Scanner t = new Scanner(System.in);

System.out.print("Enter the no. of rows: ");

int r = t.nextInt();

System.out.print("Enter the no. of columns: ");

int c = t.nextInt();

int[][] a1 = new int[r][c];

int[][] a2 = new int[r][c];

System.out.println("Enter the values of matrix 1:");

// Input values for matrix 1

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

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

a1[i][j] = t.nextInt();

}

}

System.out.println("Enter the values of matrix 2:");

// Input values for matrix 2

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

for (int l = 0; l < c; l++) {

a2[k][l] = t.nextInt();

}

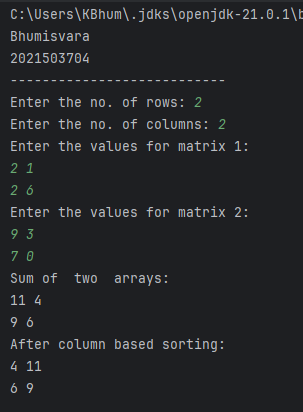
}

colSumMax(a1, a2, r, c);

}

}

**OUTPUT**



**RESULT:**

Arrays has been implemented successfully.