Insert at position traverse

import java.util.Arrays;

public class Addspecificpos{

    public static void main(String[] args) {

        int [] arr = {23,26,29,32,38};

        System.out.println("Elements before : " +Arrays.toString(arr));

        int val,pos,n,i;

        val =35;

        pos=2;

        n=arr.length;

        for( i= n-1 ; i > pos ; i--)

        {

            arr[i] = arr[i-1];

        }

        arr[pos]=val;

        System.out.println("After insertion   : " +Arrays.toString(arr));

    }

}

Elements before : [23, 26, 29, 32, 38]

After insertion   : [23, 26, 35, 29, 32]

import java.util.Arrays;

public class CopyArray{

    public static void main(String[] args) {

        int[] arr4 = {12,23,45,67,98,76,54,34};

        int n=arr4.length;

        int[] arrcopy = new int[n];

        System.out.println("Initial Array is : " +Arrays.toString(arr4));

        for(int i=0;i<n;i++)

        {

            arrcopy[i] = arr4[i];

       }

       System.out.println("After Copied : " +Arrays.toString(arrcopy));

    }

}

Initial Array is : [12, 23, 45, 67, 98, 76, 54, 34]

After Copied : [12, 23, 45, 67, 98, 76, 54, 34]

import java.util.Arrays;

import java.lang.Math;

public class DuplArray{

    public static void main(String[] args) {

        int[] arr = {23,45,76,54,23,87,65,2,346,778,98};

        int n1=arr.length;

        int[] arr2 = {1,18,543,234,45,58,98,65,98};

        int n2=arr2.length;

        int[] arr3= new int[Math.min(n1, n2)];

        int count = 0;

**for(int i=0;i<n1;i++){**

**for(int j=0;j<n2;j++){**

**if(arr[i] == arr2[j]){**

**// u can simply print here.no need any count.this is for ur knowledge.**

**arr3[count]=arr[i];**

**count++;**

                }

            }

        }

        /\*

         \* So, Arrays.copyOf(arr3, count) creates a new array with the exact size

         \* needed to hold all the common elements found.

         \* The reference to this new array is then assigned back to the variable arr3,

         \* effectively updating it with the properly sized array.  \*/

        arr3=Arrays.copyOf(arr3, count);

        System.out.println(Arrays.toString(arr3));

    }    }

   // Import the Arrays class from the java.util package.

   import java.util.Arrays;

public class Duplarr2{

    // The main method where the program execution starts.

    public static void main(String[] args) {

        // Declare and initialize an integer array 'my\_array'.

        int[] my\_array = {1, 2, 5, 5, 6, 6, 7, 2};

        // Iterate through the elements of the array.

        for (int i = 0; i < my\_array.length-1; i++) {

            for (int j = i+1; j < my\_array.length; j++) {

                // Check if two elements are equal and not the same element.

                /\*

                 \* i != j: This part ensures that the duplicate elements found are not the same element at the same position.

                 \*  If i is equal to j, it means the elements are the same, and it doesn't consider it as a duplicate.

                 \*/

                if ((my\_array[i] == my\_array[j]) && (i != j)) {

                    // If a duplicate is found, print the duplicate element.

                    System.out.println("Duplicate Element : " + my\_array[j]);

                }

            }

        }

    }

}

Duplicate Element : 2

Duplicate Element : 5

Duplicate Element : 6

public class Felement{

    public static void findElement(int[] arr2,int value){

        int n = arr2.length;

        boolean found=false;

          for(int i=0;i<n;i++)

            {

                if(arr2[i] == value)

                {

                    System.out.println("ELEMENT FOUND IN " +(i+1) +" INDEX" );

                    found=true;

                    break;

                }

            }

            if(!found)

                {

                    System.out.println("NOT FOUND !!!!!!!!");

                }

        }

    public static void main(String[] args) {

        int[] arr2 = {12,34,56,78,98,9,87,65};

        Felement.findElement(arr2,56);

    }

}

public class Flarge {

    public static void largest(int[] arr){

        int max = 0;

        max=arr[0];

        for(int i=0;i<arr.length;i++)

        {

            if (arr[i] > max)

            {

                max=arr[i];

            }

        }

        System.out.println("max is :" +max);

    }

    public static void main(String[] args) {

        int[] arr3 = {12,3,45,56,78,90,123,23,100,5};

        Flarge.largest(arr3);

    }

}

import java.util.Arrays;

public class Exer2 {

    public static void main(String[] args) {

        int[] array = {78, 89, 90, 12, 65};

        int length = array.length;

        boolean check = false;

        int find = 90;

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

            if (array[i] == find) {

                System.out.println("Index found in position: " + (i + 1));

                check = true;

                break;

            }

        }

        if (!check) {

            System.out.println("Index not found!!");

        }

        int pos=3,value=45;

        for(int i=pos-1;i > pos; i--)

        {

            array[i]=array[i-1];

        }

        array[pos]=value;

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

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

        {

            if(array[i]==89)

            {

                System.out.println(i);

            }

        }

        for(int i = 1;i < array.length-1 ; i++)

        {

            array[i]=array[i+1];

        }

        array=Arrays.copyOf(array, 4);

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

    }

}

import java.util.Arrays;

import java.util.Collections;

public class Rreverse {

    public static void main(String[] args) {

        Integer[] arr = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

        System.out.println("Original array: " + Arrays.toString(arr));

        // Convert array to List, reverse it, and convert back to array

        Collections.reverse(Arrays.asList(arr));

        System.out.println("Reversed array: " + Arrays.toString(arr));

    }

}

import java.util.Arrays;

public class Sortt{

    public static void main(String[] args) {

        int[] arr = {2,4,1,3,7,6,8,5,9,10};

        System.out.println("Array before sort : " +Arrays.toString(arr));

        System.out.println();

        Arrays.sort(arr);

        System.out.println("Array after sort : " +Arrays.toString(arr));

        System.out.println("Second largest Array: " +arr[1]);

    }

}

import java.util.Arrays;

public class Revmanual{

public static void main(String[] args) {

int[] arr = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

System.out.println("Original array: " + Arrays.toString(arr));

int start = 0;

int end = arr.length - 1;

while (start < end) {

// Swap elements at start and end indices

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

// Move towards the center

start++;

end--;

}

System.out.println("Reversed array: " + Arrays.toString(arr));

System.out.println();

int n=arr.length;

int[] arr3=new int[n];

int j=0;

for(int i=n-1;i>-1;i--)

{

arr3[j]=arr[i];

j++;

}

System.out.println(Arrays.toString(arr3));

}

}

import java.util.Arrays;

import java.util.Scanner;

public class Rremove{

    public static void main(String[] args) {

        int[] arr = {23,45,75,23,98,65,34,76,34};

        Scanner sc = new Scanner(System.in);

        int n = arr.length;

        System.out.println("Array before removing duplicates : " +Arrays.toString(arr));

        System.out.println("Enter element index to remove (first element is = 0):" );

        int index = sc.nextInt();

        if( index < n && index > 0)

        {

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

            {

                arr[i]=arr[i+1];

            }

        }

        else

        {

            System.out.println("Index out of bound");

        }

        arr=Arrays.copyOf(arr, n-1);

        System.out.println("After deletion : " +Arrays.toString(arr));}}

import java.util.Scanner;

public class StudentDatabase{

    private static final int MAX\_STUDENTS = 50;

    private String[] studentNames;

    private int[] studentNumbers;

    private int currentSize;

    public StudentDatabase() {

        studentNames = new String[MAX\_STUDENTS];

        studentNumbers = new int[MAX\_STUDENTS];

        currentSize = 0;

    }

    public void insertStudent(int studentNumber, String studentName) {

        if (currentSize < MAX\_STUDENTS) {

            studentNumbers[currentSize] = studentNumber;

            studentNames[currentSize] = studentName;

            currentSize++;

            System.out.println("Student inserted successfully.");

        } else {

            System.out.println("Student database is full. Cannot insert more students.");

        }

    }

    public void searchStudent(int studentNumber) {

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

            if (studentNumbers[i] == studentNumber) {

                System.out.println("Student found: " + studentNames[i]);

                return;

            }

        }

        System.out.println("Student not found with number: " + studentNumber);

    }

    public void deleteStudent(int studentNumber) {

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

            if (studentNumbers[i] == studentNumber) {

                System.out.println("Student deleted: " + studentNames[i]);

                // Move the last student information to the deleted position

                studentNumbers[i] = studentNumbers[currentSize - 1];

                studentNames[i] = studentNames[currentSize - 1];

                currentSize--;

                return;

            }

        }

        System.out.println("Student not found with number: " + studentNumber);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        StudentDatabase studentDatabase = new StudentDatabase();

        while (true) {

            System.out.println("Choose an action:");

            System.out.println("1. Insert a student");

            System.out.println("2. Search for a student");

            System.out.println("3. Delete a student");

            System.out.println("4. Exit");

            int choice = scanner.nextInt();

            scanner.nextLine(); // Consume the newline character

            switch (choice) {

                case 1:

                    System.out.print("Enter student number: ");

                    int studentNumber = scanner.nextInt();

                    scanner.nextLine(); // Consume the newline character

                    System.out.print("Enter student name: ");

                    String studentName = scanner.nextLine();

                    studentDatabase.insertStudent(studentNumber, studentName);

                    break;

                case 2:

                    System.out.print("Enter student number to search: ");

                    int searchNumber = scanner.nextInt();

                    scanner.nextLine(); // Consume the newline character

                    studentDatabase.searchStudent(searchNumber);

                    break;

                case 3:

                    System.out.print("Enter student number to delete: ");

                    int deleteNumber = scanner.nextInt();

                    scanner.nextLine(); // Consume the newline character

                    studentDatabase.deleteStudent(deleteNumber);

                    break;

                case 4:

                    System.out.println("Exiting program.");

                    System.exit(0);

                    break;

                default:

                    System.out.println("Invalid choice. Please enter a valid option.");

            }

        }

    }

}