PROGRAMS ON CALL BY REFERENCE

Question:

A class Mixer has been defined to merge two sorted integer arrays in ascending order. Some of the members of the class are given below:

**Class name                         :**Mixer

**Data members/instance variables:**

int arr[]                                 :               to store the elements of an array  
int n                                      :              to store the size of the array

**Member functions:**

Mixer( int nn)                     :               constructor to assign n = nn  
void accept()                     :               to accept the elements of the array in ascending order without any duplicates  
Mixer mix( Mixer A)             :              to merge the current object array elements with the parameterized array elements and return the resultant object  
void display()                    :               to display the elements of the array

Specify the class **Mixer**, giving details of the **constructor(int)**, **void accept()**, **Mixer mix(Mixer)** and**void display()**. Define the **main()** function to create an object and call the function accordingly to enable the task.

import java.io.\*;

class Mixer

{

    int arr[];

    int n;

    static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    Mixer(int nn)

    {

        n = nn;

        arr = new int[n];

    }

    void accept()throws IOException

    {

        System.out.println("\n\* Input the Array \*\n");

        for(int i=0; i<n; i++)

        {

            System.out.print("Enter Element ["+(i+1)+"] : ");

            arr[i] = Integer.parseInt(br.readLine());

        }

        System.out.println();

    }

    Mixer mix(Mixer A)

    {

        int size = this.arr.length + A.arr.length; //size of resulting array

        /\* 'this' keyword refers to the current object, i.e. the object which calls mix() function \*/

        Mixer B = new Mixer(size); //object which will store the result of merging

        int x = 0;

        /\* Merging the array of current object with array of parameter object \*/

        for(int i=0; i<size; i++)

        {

            if(i<A.arr.length)

                B.arr[i] = A.arr[i];

            else

            {

                B.arr[i] = this.arr[x];

                x++;

            }

        }

        /\* Sorting the result\*/

        int temp=0;

        for(int i=0; i<size-1; i++)

        {

            for(int j=i+1; j<size; j++)

            {

                if(B.arr[i]>B.arr[j])

                {

                    temp = B.arr[i];

                    B.arr[i] = B.arr[j];

                    B.arr[j] = temp;

                }

            }

        }

        return B;

    }

    void display()

    {

        for(int i=0; i<n; i++)

        {

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

        }

        System.out.println();

    }

    public static void main(String args[])throws IOException

    {

        System.out.print("Enter size of the 1st array : ");

        int p = Integer.parseInt(br.readLine());

        Mixer obj1 = new Mixer(p);

        obj1.accept();

        System.out.print("Enter size of the 2nd array : ");

        int q = Integer.parseInt(br.readLine());

        Mixer obj2 = new Mixer(q);

        obj2.accept();

        Mixer obj3 = obj2.mix(obj1); //obj2 is the current object which is referred by 'this' keyword above

        System.out.print("The 1st Array is : ");

        obj1.display();

        System.out.print("The 2nd Array is : ");

        obj2.display();

        System.out.print("The Merged Array is : ");

        obj3.display();

    }

}





