1.WAP to find the duplicates present in an arrayP to find the duplicates present in an array

Program:-

import java.util.\*;

class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int a[]=new int[10];

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

{

a[i]=s.nextInt();

}

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

{

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

{

if(a[i]==a[j])

{

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

}

}

}

}

}

Output:-

Text

Description automatically generated

2. WAP to sort an array using Quick Sort Algorithm.

import java.util.\*;

class Main

{

int partition(int arr[], int low, int high)

{

int pivot = arr[high];

int i = (low-1);

for (int j=low; j<high; j++)

{

if (arr[j] <= pivot)

{

i++;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int temp = arr[i+1];

arr[i+1] = arr[high];

arr[high] = temp;

return i+1;

}

void sort(int arr[], int low, int high)

{

if (low < high)

{

int pi = partition(arr, low, high);

sort(arr, low, pi-1);

sort(arr, pi+1, high);

}

}

static void printArray(int arr[])

{

int n = arr.length;

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

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

System.out.println();

}

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

int arr[]=new int[10];

for(int i=0;i<10;i++)

{

arr[i]=s.nextInt();

}int n = arr.length;

Main ob = new Main();

ob.sort(arr, 0, n-1);

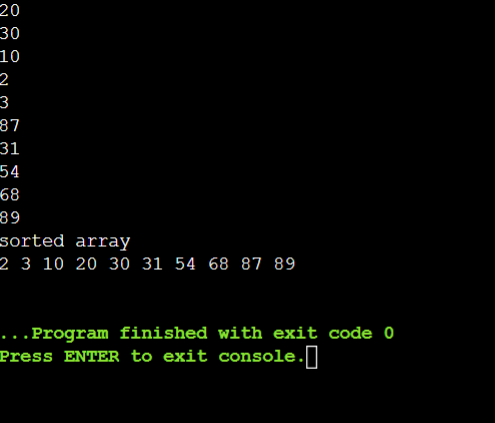
System.out.println("sorted array");

printArray(arr);

}

}

Output:-



3. WAP to sort an array using Merge Sort Algorithm.

Program:-

class Main {

void merge(int arr[], int l, int m, int r)

{

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

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

L[i] = arr[l + i];

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

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

int i = 0, j = 0;

int k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

void sort(int arr[], int l, int r)

{

if (l < r) {

int m = l + (r - l) / 2;

sort(arr, l, m);

sort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

static void printArray(int arr[])

{

int n = arr.length;

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

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

System.out.println();

}

public static void main(String args[])

{

int arr[] = { 12, 11, 13, 5, 6, 7 };

System.out.println("Given Array");

printArray(arr);

Main ob = new Main();

ob.sort(arr, 0, arr.length - 1);

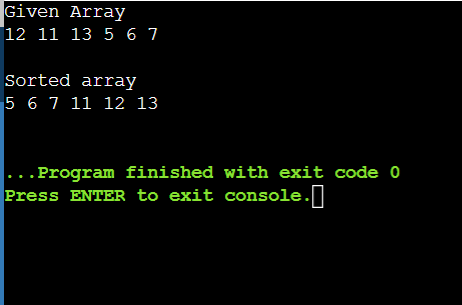
System.out.println("\nSorted array");

printArray(arr);

}

}

Output:-



4. WAP to sort an array using Bubble Sort Algorithm

Program:-

import java.util.\*;

class Main {

void bubbleSort(int 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]) {

int temp = arr[j];

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

arr[j + 1] = temp;

}

}

void printArray(int arr[])

{

int n = arr.length;

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

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

System.out.println();

}

public static void main(String args[])

{

Main ob = new Main();

int arr[] = { 5, 1, 4, 2, 8 };

ob.bubbleSort(arr);

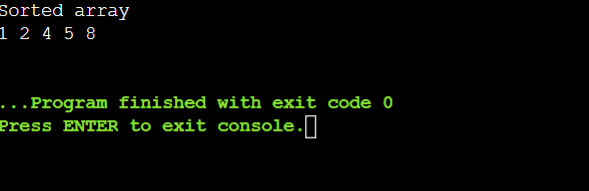
System.out.println("Sorted array");

ob.printArray(arr);

}

}

Output:-



5. WAP to sort an array using Selection Sort Algorithm

import java.util.\*;

class Main{

void sort(int arr[])

{

int n = arr.length;

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

{

int min\_idx = i;

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

if (arr[j] < arr[min\_idx])

min\_idx = j;

int temp = arr[min\_idx];

arr[min\_idx] = arr[i];

arr[i] = temp;

}

}

void printArray(int arr[])

{

int n = arr.length;

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

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

System.out.println();

}

public static void main(String args[])

{

Main ob = new Main();

int arr[] = {64,25,12,22,11};

ob.sort(arr);

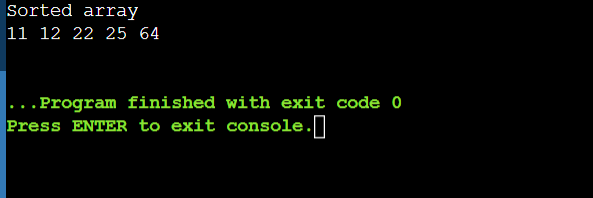
System.out.println("Sorted array");

ob.printArray(arr);

}

}

Output:-



6. WAP to check whether an array is a subset of another array.

Program:-

class Main {

static boolean isSubset(int arr1[], int arr2[], int m,

int n)

{

int i = 0;

int j = 0;

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

for (j = 0; j < m; j++)

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

break;

if (j == m)

return false;

}

return true;

}

public static void main(String args[])

{

int arr1[] = { 11, 1, 13, 21, 3, 7 };

int arr2[] = { 11, 3, 7, 1 };

int m = arr1.length;

int n = arr2.length;

if (isSubset(arr1, arr2, m, n))

System.out.print("arr2[] is "

+ "subset of arr1[] ");

else

System.out.print("arr2[] is "

+ "not a subset of arr1[]");

}

}

Output:-

