ASSIGNMENT 2

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

```
public class Duplicates {
public static void main(String[] args) {
int [] arr = new int [] {1, 2, 3, 3, 2, 9, 8, 9, 3};

System.out.println("Duplicate elements in array: ");

 for(int i = 0; i < arr.length; i++)
 {
    for(int j = i + 1; j < arr.length; j++)
    {
      if(arr[i] == arr[j])
      System.out.println(arr[j]);
            }
        }
    }
    }
}
```

---

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

```
import java.util.Scanner;

public class Quicksort
{
public static int partition(int a[],int l,int h){
int i=l+1 ,j=h,c=l,temp;

    for(; i<=j ;)
    {

        while(i<=h && a[i]<a[c] )
```

```
            i++;
        while(a[j]>a[c] && j>l)
            j--;

        if(i<j)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
        else
        break;
        }


    temp=a[c];
    a[c]=a[j];
    a[j]=temp;
    return j;
    }

    public static void Sort(int a[],int l,int h)
    {
        if(l<h)
        {
            int m=partition(a,l,h);
            Sort(a,l,m-1);
            Sort(a,m+1,h);

        }

    }

public static void parray(int a[])
{
for(int i=0; i < a.length; i++)
{
```

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

}
public static void main(String[] args)
{
    int n, res,i;
    Scanner s = new Scanner(System.in);
    System.out.print("Enter number of elements:");
    n = s.nextInt();
    int a[] = new int[n];
System.out.println("Enter "+n+" elements ");
    for( i=0; i < n; i++)
    {
    a[i] = s.nextInt();
    }

System.out.println( "Elements in array ");
parray(a);
Sort(a,0,n-1);
System.out.println( " after sorting");
parray(a);

    }

}
```

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

```
import java.util.Arrays;

public class Bubblesort {

    public static void main(String[] args) {
        int a[]= {4,6,2,1,8,9};
        int n=a.length;
        System.out.println("Initial
array:"+Arrays.toString(a));
        for(int i=0;i<n-1;i++)
        {

            for(int j=0;j<n-1;j++)
            {
                if(a[j] > a[j+1])
                {
                    int temp=a[j];
                    a[j]=a[j+1];
                    a[j+1]=temp;

                }
            }

    }
        System.out.println("Sorted array
is:"+Arrays.toString(a));
}
}
```

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

```
import java.util.Scanner;
public class Mergesort
{

public static void merge(int a[],int l,int m,int h){
       int i, j,c=l;
       int b[]=new int[h+1];

    for(i = l,j = m+1; i<=m && j<=h; c++)
             {

            if(a[i] <= a[j])
            b[c] = a[i++];
         else
            b[c] = a[j++];
        }
     while(i <= m )
                 b[c++] = a[i++];

               while(j<=h)
                 b[c++] = a[j++];

     for(i = l ; i <= h; i++)
                        a[i] = b[i];
   }

   public static void Sort(int a[],int l,int h)
   {
         if(l<h)
        {
           int m=(l+h)/2;
           Sort(a,l,m);
           Sort(a,m+1,h);
           merge(a,l,m,h);

         }

   }
 public static void printarray(int a[])
{
       for(int i=0; i < a.length; i++)
        {

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

}
    public static void main(String[] args)
    {
        int n, res,i;
        Scanner s = new Scanner(System.in);
        System.out.print("Array size:");
        n = s.nextInt();
        int a[] = new int[n];
        System.out.println("Enter "+n+" elements ");
        for( i=0; i < n; i++)
        {
            a[i] = s.nextInt();
        }

        System.out.println( "Elements in array ");
        printarray(a);
         Sort(a,0,n-1);
        System.out.println( " after sorting");
         printarray(a);
       }
}
```

5.WAP to sort an array using Selection Sort Algorithm.

```
public class SelectionSort {

    public static void main(String[] args) {
        int arr[]= {10,5,6,7,1,9};
        int length=arr.length;
        for(int i=0;i<length-1;i++)
        {
            int min_index=i;
            for(int j=i+1;j<length;j++)
            {
            if(arr[min_index] > arr[j])
                min_index=j;
            }
            int temp=arr[min_index];
            arr[min_index] = arr[i];
            arr[i]=temp;
        }
        System.out.println("Sorted array:");
        for(int i=0;i<length;i++)

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

    }

}
```

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

```
import java.util.Scanner;
class Subset
{
 public static void main(String ar[])
  {
   int n,m;
   Scanner sc=new Scanner(System.in);

   System.out.println("Enter Array1 size:");
   n=sc.nextInt();

   System.out.println("Enter  array2 size ");
   m=sc.nextInt();
   //arays
   int a[]=new int[n];
   int b[]=new int[m];
   System.out.println("Enter array1:");
   for(int i=0;i<n;i++)
     a[i]=sc.nextInt();
   int j=0;
   System.out.println("Enter array2");
   for(j=0;j<m;j++)
      b[j]=sc.nextInt();
   int x=0;
   for(int i=0;i<m;i++)
    {
      for(j=0;j<n;j++)
      {
        if(b[i]==a[j])
        {
         break;
        }
      }

      if(j==n)
      {
      x=1;
      System.out.println("Given array is not a 
subset");
       break;
      }
    }

  if(x==0)
  {
    System.out.println("Given array is a subset of the 
main array");
  }
 }
}
```