**DAY 9-ASSIGNMENT**

**1.**

**package** Sai;  
public class QuickSort {    
    
   public static void quick (int[] a, int low, int high) {    
       if (low < high) {    
           int pi = partition(a, low, high);    
           quickSort(a, low, pi - 1);    
           quickSort(a, pi + 1, high);    
       }    
   }    
    
   public static int partition(int[] a, int low, int high) {    
       int pivot = a[high];    
       int i = (low - 1);    
       for (int j = low; j < high; j++) {    
           if (a[j] < pivot) {    
               i++;    
               int temp = a[i];    
               a[i] = a[j];    
               a[j] = temp;    
           }    
       }    
       int temp = a[i + 1];    
       a[i + 1] = a[high];    
       a[high] = temp;    
       return i + 1;    
   }    
   public static void main(String[] args) {    
       Scanner sc= new Scanner(System.in);    
       System.out.println("Enter no of elements: ");    
       int n = sc.nextInt();    
       int[] a = new int[n];        
       System.out.println("Enter array elements:");    
       for (int i = 0; i < n; i++) {    
           a[i] = sc.nextInt();    
       }    
       quickSort(a, 0, a.length - 1);    
       System.out.println("Sorted array elements:");    
       for (int num : a) {    
           System.out.print(num + " ");    
       }    
       sc.close();    
   }    
}

**2.**

**package** Sai**;**

**import** java.util.Scanner;

**public** **class** Merge {

**public** **static** **void** merge(**int**[] a, **int** left, **int** middle, **int** right) {

**int** n1 = middle - left + 1;

**int** n2 = right - middle;

**int**[] L = **new** **int**[n1];

**int**[] R = **new** **int**[n2];

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

           L[i] = a[left + i];

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

           R[j] = a[middle + 1 + j];

**int** i = 0, j = 0;

**int** k = left;

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

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

               a[k] = L[i];

               i++;

           } **else** {

               a[k] = R[j];

               j++;

           }

           k++;

       }

**while** (i < n1) {

           a[k] = L[i];

           i++;

           k++;

       }

**while** (j < n2) {

           a[k] = R[j];

           j++;

           k++;

       }

   }

**public** **static** **void** sort(**int**[] a, **int** left, **int** right) {

**if** (left < right) {

**int** middle = (left + right) / 2;

*sort*(a, left, middle);

*sort*(a, middle + 1, right);

*merge*(a, left, middle, right);

       }

   }

**public** **static** **void** main(String[] args) {

       Scanner sc = **new** Sc(System.***in***);

       System.***out***.println("Enter no of array elements:");

**int** n = sc.nextInt();

**int**[] a= **new** **int**[n];

       System.***out***.println("Enter array elements:");

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

           a[i] = sc.nextInt();

       }

*sort*(a, 0, a.length - 1);

       System.***out***.println("Sorted array elements:");

**for** (**int** num : a) {

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

       }

       sc.close();

   }

}

**3.**

**package** Sai;

**import** java.util.Scanner;

**public** **class** Linear{

**public** **static** **void** main(String[] args) {

       Scanner sc = **new** Scanner(System.***in***);

       System.***out***.print("Enter the number of elements in the array: ");

**int** n = sc.nextInt();

**int**[] a = **new** **int**[n];

       System.***out***.print("Enter the elements of the array: ");

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

           a[i] = sc.nextInt();

       }

       System.***out***.print("Enter the element to search for: ");

**int** x = sc.nextInt();

**boolean** found = **false**;

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

**if** (a[i] == x) {

               found = **true**;

**break**;

           }

       }

**if** (found) {

           System.***out***.println("Element is Present");

       } **else** {

           System.***out***.println("Element is not Present");

       }

       sc.close();

   }

}

**4.**

**package** Sai;

**public** **class** Selection{

**public** **static** **void** main(String[] args) {

       String[] fruits = {"banana", "apple", "orange", "grape", "kiwi"};

*selectionSort*(fruits);

**for**(String fruit:fruits ) {

         System.***out***.print(fruit+" ");

       }

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

**for** (String fruit : fruits) {

           System.***out***.print(fruit + " ");

       }

   }

**public** **static** **void** selectionSort(String[] arr) {

**int** n = a.length;

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

**int** minIndex = i;

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

**if** (a[j].compareTo(arr[minIndex]) < 0) {

                   minIndex = j;

               }

           }

           String temp = arr[minIndex];

           a[minIndex] = a[i];

           a[i] = temp;

       }

   }

} 

**5.**

**package** Sai;

**public** **class** InsertionSort{

**public** **static** **void** main (String [] args) {

       Scanner sc= **new** Scanner (System.***in***);

       System.***out***.print("Enter the number of floating-point numbers: ");

**int** n = sc. nextInt ();

**double**[] a = **new** **double**[n];

       System.***out***.println("Enter the floating-point numbers: ");

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

           a[i] = sc.nextDouble();

       }

*insertionSort*(arr);

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

**for** (**double** num : a) {

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

       }

       sc.close();

   }

**public** **static** **void** insertionSort(**double**[] arr) {

**int** n = a.length;

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

**double** key = a[i];

**int** j = i - 1;

**while** (j >= 0 && a[j] > key) {

               a[j + 1] = a[j];

               j = j - 1;

           }

           a[j + 1] = key;

       }

   }

}

**6.**   
**package** Sai;  
public class BinarySearch{    
   public static void main(String[] args) {    
       Scanner sc = new Scanner(System.in);    
    
       System.out.print("Enter the strings (space-separated): ");    
       String input = sc.nextLine();    
       String[] a = input.split(" ");    
    
       System.out.print("Enter the string to search for: ");    
       String target = sc.nextLine();    
    
       Arrays.sort(a);    
    
       int result = binarySearch(a, target);    
    
       if (result == -1) {    
           System.out.println("Element " + target + " not found");    
       } else {    
           System.out.println("Element " + target + " found at index " + result);    
       }    
       sc.close();    
   }    
    
   public static int binarySearch(String[] a, String target) {    
       int left = 0, right = a.length - 1;    
       while (left <= right) {    
           int mid = left + (right - left) / 2;    
           int res = target.compareTo(arr[mid]);    
    
           if (res == 0) {    
               return mid;    
           }    
    
           if (res > 0) {    
               left = mid + 1;    
           }    
    
           else {    
               right = mid - 1;    
           }    
       }    
    
       return -1;    
   }    
}