

# Preparazione Esame

PROVA B

# Esercizio 1

Scrivi una classe `ListaOrdinataInt` (spiegata dopo) con package uguale a `cognome_nome` nel file `ListaOrdinataInt.java`. **`ListaOrdinataInt`** rappresenta una collezione di interi gestiti in modo che siano in ordine. E' una struttura dinamica ma non è generica.

Ha un metodo **`insInt`** che prende un intero e lo inserisce in ordine.

Ha un metodo **`find`** che dato un intero restituisce `true` se quell'intero è nella lista, altrimenti `false`. Se la lista è vuota solleva eccezione.

Implementa anche un **`toString()`** che puoi usare per testare la tua classe.

In questa classe scrivi anche un **`main`** di prova in cui provi la tua struttura dati. Non fare alcuni input con l'utente. Alla fine consegnerai un solo file `ListaOrdinataInt.java`

# Esercizio 2

Scrivi la definizione di albero binario di ricerca generico.

Definisci un metodo per inserire gli elementi in ordine e un metodo per stampare gli elementi inseriti. Scrivi un esempio di uso.

# Esercizio 3

Scrivi un metodo statico che dato un array di interi (int) stampa alla fine il numero massimo e la media, e la mediana. Scrivi anche un esempio di utilizzo.

# Esercizio 4

Scrivi l'algoritmo di mergesort per un array di double.

La classe MergeSorterDouble effettua l'ordinamento in ordine decrescente con un metodo da implementare (statico).

Scrivi anche un main di prova all'interno della classe stessa

```

package paolo_vavassori;

import java.util.Arrays;

public class MergerAndSorter {

    private static double[] merge(double[] a, double[] b) {
        double[] c = new double[a.length + b.length];
        int i = 0, j = 0;
        for (int k = 0; k < c.length; k++) {
            if (i >= a.length) c[k] = b[j++];
            else if (j >= b.length) c[k] = a[i++];
            else if (a[i] >= b[j]) c[k] = a[i++];
            else c[k] = b[j++];
        }
        return c;
    }

    public static double[] mergesort(double[] input) {
        int N = input.length;
        if (N <= 1) return input;
        double[] a = new double[N/2];
        double[] b = new double[N - N/2];
        for (int i = 0; i < a.length; i++) a[i] = input[i];
        for (int i = 0; i < b.length; i++) b[i] = input[i + N/2];
        return merge(mergesort(a), mergesort(b));
    }

    public static void main(String[] args) {

        double[] original = { 45, 4, 89, 90, 3, 2, 1, 34 ,100};
        MergerAndSorter.mergesort(original);
        System.out.println(Arrays.toString(MergerAndSorter.mergesort(original)));
    }
}

```