

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

## Appello del 2 luglio 2019

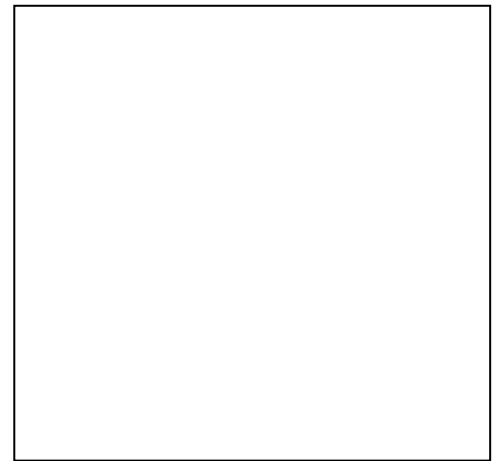
### Esercizio 1 (4 punti)

Si scriva nel riquadro cosa stampa il seguente programma Java:

```
public class MainClass {
    public static void main(String[] args) {
        int[] a = {10, 2, 20, 10, 9, 10, 2};
        System.out.println(enigma(a));
    }

    static int enigma(int[] a) {
        int ris = 0, i=0;
        while (ris<2 && i<a.length) {
            int conto=0;
            for (int j=0; j<i; j++) {
                if (a[j] < a[i]) {
                    conto++;
                }
            }
            if (conto>1) {
                System.out.println
                    ("a[" + i + "]=" + a[i]);
                ris++;
            }

            i++;
        }
        return ris;
    }
}
```



### Esercizio 2 (12 punti)

L'array delle ripetizioni di un dato array **a** di numeri interi è definito come un array avente come lunghezza il numero di elementi distinti dell'array **a**, e contenente in posizione **i** ( $i=0,1,\dots$ ) il numero di ripetizioni nell'array **a** dell' $(i+1)$ -esimo elemento più piccolo di **a**. Ad esempio, se  $a=\{8, 1, 3, 1, 5, 10, 1, 3\}$ , il suo array di ripetizioni è  $\{3,2,1,1,1\}$  in quanto in **a** ci sono 5 elementi distinti, e l'elemento più piccolo di **a** (1) compare 3 volte, il secondo elemento più piccolo (3) compare 2 volte, e i rimanenti elementi compaiono solo una volta.

Scrivere un metodo in Java

```
static int[] ripetizioni (int[] a)
```

che, preso come parametro un array **a** di numeri interi, crea e restituisce il suo array delle ripetizioni.

Se **a** vale *null*, viene restituito *null*; se **a** è vuoto, viene restituito un array vuoto.

**Si analizzi la complessità temporale del metodo proposto:** tale metodo può richiamare qualsiasi algoritmo di ordinamento e/o di ricerca visto a lezione e deve avere complessità temporale  **$O(n \log n)$**  nel caso peggiore (soluzioni con complessità temporale peggiore danno luogo a una valutazione minore, pari a un massimo di 5 punti totali), dove **n** è la lunghezza dell'array **a** in input.

---

### Esercizio 3 (11 punti)

Si consideri il seguente array di numeri interi:

**{10, 20, 5, 50, 31, 14, 2, 6, 1, 8, 25}**

Mostrare **passo-passo l'esecuzione del merge-sort** per ordinare l'array in modo non decrescente.

Si descriva la complessità computazionale del merge-sort (anche scrivendo e risolvendo la ricorrenza  $T(n)$ ).

---

### Esercizio 4 (8 punti)

Si consideri il seguente tipo riferimento definiti dall'utente

```
class Data {  
    int giorno;  
    int mese;  
    int anno;  
}
```

- **(4 punti)** Scrivere un metodo **static boolean dataValida (Data d)** che, preso come parametro una data **d**, restituisce true se e solo se la data è valida (si tenga in considerazione anche che gli anni bisestili sono quelli divisibili per 4). Il metodo deve gestire opportunamente tutti i casi in cui le variabili di tipo riferimento coinvolte valgano *null*.
- **(4 punti)** Scrivere un metodo **static Data minimo (Data[] aDate)** che, preso come parametro un array di Data **aDate**, restituisce la data minima tra le date valide presenti nell'array. Il metodo deve gestire opportunamente tutti i casi in cui le variabili di tipo riferimento coinvolte valgano *null*.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **100** minuti.
- Scrivere **subito** nome, cognome, matricola su **OGNI FOGLIO (compreso questo)**.
- Le risposte al primo esercizio devono essere date direttamente nel riquadro di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato e depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.