

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

Matricola \_\_\_\_\_

---

## Appello del 22 dicembre 2016

---

### Esercizio 1 (5 punti)

Scrivere un metodo

**public static int contaOccorrenze (int[] a, int n)**

che, preso come parametro un array *a* di numeri interi e un intero *n*, senza modificare il contenuto dell'array, restituisce il numero di occorrenze dell'intero *n* all'interno dell'array *a*. Se *a* vale *null*, viene restituito 0.

Ad esempio, se *a* = [10, 6, 5, 4, 5, 6] e *n*=5 deve essere restituito 2.

---

### Esercizio 2 (5 punti)

Scrivere un metodo

**public static boolean primaOccorrenza (int[] a, int pos)**

che, preso come parametro un array di numeri interi *a* e un intero *pos*, senza modificare il contenuto dell'array, restituisce *true* se e solo se l'elemento in posizione *pos* non compare nell'array *a* nelle posizioni da 0 a *pos*-1.

Se *a* vale *null*, viene restituito *false*. Se *pos* è fuori dagli indici ammissibili per l'array, viene restituito *false*.

Ad esempio, se *a*={5,3,7,7,3,10,5} e *pos* = 2, il metodo deve restituire *true* in quanto l'elemento 7 compare per la prima volta in posizione 2.

---

### Esercizio 3 (6 punti)

Scrivere un metodo

**public static int[] multiIntersezione (int[] a, int[] b)**

che, presi come parametri due array di numeri interi *a* e *b*, senza modificare il contenuto degli array, crea e restituisce un nuovo array contenente l'intersezione dei multisinsiemi rappresentati dagli array *a* e *b*.

Nell'intersezione di due multisinsiemi *a* e *b* compaiono tutti e soli gli elementi comuni ai multisinsiemi con un numero di ripetizioni pari al minimo tra il numero di ripetizioni dell'elemento in *a* e quello in *b*.

Se *a* o *b* valgono *null*, viene restituito *null*.

Ad esempio, se *a*={5, 3, 7, 7, 3, 10, 5} e *b*={20, 5, 7, 5, 30}, il metodo deve restituire {5, 5, 7}.

Il metodo **deve** richiamare i metodi *contaOccorrenze* e *primaOccorrenza* e potrà applicare il seguente algoritmo:

- prima bisogna contare quanti elementi vanno inseriti nell'array da restituire:
  - per ogni posizione *i* dell'array *a*, se *i* è la prima occorrenza dell'elemento *a*[*i*], devono essere contati un numero di elementi in numero uguale al minimo tra le occorrenze di *a*[*i*] in *a* e le occorrenze di *a*[*i*] in *b*
- bisogna creare l'array da restituire dell'opportuna lunghezza
- bisogna infine riempire l'array creato:
  - per ogni posizione *i* dell'array *a*, se *i* è la prima occorrenza dell'elemento *a*[*i*], devono essere inseriti un numero di elementi uguali ad *a*[*i*] in numero uguale al minimo tra le occorrenze di *a*[*i*] in *a* e le occorrenze di *a*[*i*] in *b*

---

#### Esercizio 4 (6 punti)

Cosa stampa il seguente programma Java?

```
public class Main {
    public static void main(String[] args) {
        System.out.println(enigma(1,4));
        System.out.println(enigma(5,-2));
        System.out.println(enigma(900,0));
    }
    static int enigma (int x, int y){
        if (x<y) return enigma (y,x);
        if (x==y) return 0;
        return 2 + enigma (x-1,y);
    }
}
```



---

#### Esercizio 5 (10 punti)

Si consideri il tipo di dato

```
class Scena {
    int id;
    int[] personaggi;
}
```

che rappresenta una scena di uno spettacolo teatrale, in cui **id** è un identificativo numerico (non negativo) della scena e **personaggi** è un array di numeri interi contenente gli identificatori (non negativi) di tutti e soli i personaggi che compaiono nella scena.

- **(5 punti)** Scrivere un metodo **static int maxIdPersonaggio (Scena[] spettacolo)** che, preso come parametro un array di Scena **spettacolo**, restituisce il massimo id di un personaggio che compare in almeno una scena dell'array **spettacolo**.  
Se **spettacolo** vale *null*, viene restituito -1.
- **(5 punti)** Scrivere un metodo **static int personaggioPiùPresente (Scena[] spettacolo)** che, preso come parametri un array di Scena **spettacolo**, restituisce l'identificatore di uno dei personaggi che compare nel maggior numero di scene.  
Se **spettacolo** vale *null*, viene restituito -1.

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

- Per svolgere il compito si hanno a disposizione **90** minuti.
- Scrivere **subito** nome, cognome, matricola e numero del compito su **OGNI FOGLIO**.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri 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
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- 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 il proprio libretto o altro documento di identità.