

Cognome: _____ Nome: _____ Matricola: _____

Esercizio A1 (20%)

Descrivere caratteristiche e modalità di gestione dei record di attivazione e della pila di attivazione.

Esercizio A2 (20%)

Scrivere, nell'ambito di una classe **Estrazione**, un metodo di classe **int[] estrai(int[] a, int val)** con la seguente specifica:

- Pre-condizione: l'array **a** è non nullo
- Post-condizione: il metodo crea e restituisce un nuovo array, che contiene un numero di elementi pari al numero di elementi di **a** uguali a **val**, in cui tutti gli elementi del nuovo array hanno valore **val** – si può anche dire che l'array restituito è formato ordinatamente da tutti gli elementi di **a** che hanno valore **val**

Ad es., se **a** è l'array di array { 8, 5, 8, 5, 8, 2, 8, 5, 8 }, allora l'invocazione **estrai(a,5)** deve creare e restituire l'array { 5, 5, 5 }, **estrai(a,2)** deve creare e restituire l'array { 2 } e **estrai(a,99)** deve creare e restituire l'array vuoto { }.

Esercizio A3 (20%)

Definire una classe **Orario** per istanziare oggetti con le seguenti caratteristiche:

- un oggetto **Orario** rappresenta un orario nell'ambito di una giornata, espresso in ore e minuti – ad esempio, l'orario 12:00 (mezzogiorno) o l'orario 23:59 (mezzanotte meno un minuto)
- va definito un costruttore che, dati due numeri interi **hh** e **mm**, con **hh** compreso tra 0 e 23 e **mm** compreso tra 0 e 59, crea uno nuovo oggetto **Orario** che rappresenta l'orario **hh:mm**
- va definito un metodo d'istanza **int ore()** che restituisce le ore di questo orario (un numero compreso tra 0 e 23)
- va definito un metodo d'istanza **int minuti()** che restituisce i minuti di questo orario (un numero compreso tra 0 e 59)
- va definito un metodo d'istanza **String toString()** che calcola una descrizione testuale di questo orario (nella forma hh:mm)
- va definito un metodo d'istanza **void avanza(int m)** che avanza questo orario di **m** minuti (**m** è un numero naturale), anche tenendo conto delle seguenti osservazioni:
 - facendo avanzare di un minuto l'orario 12:14 si ottiene l'orario 12:15
 - facendo avanzare di tre minuti l'orario 12:58 si ottiene l'orario 13:01
 - facendo avanzare di sessantadue minuti l'orario 23:59 si ottiene l'orario 01:01

Ad esempio:

```
Orario a,b;
a = new Orario(12,0);           // a è mezzogiorno
b = new Orario(23,59);         // b è mezzanotte meno un minuto
System.out.println(a.ore());   // visualizza 12
System.out.println(a.minuti()); // visualizza 0
System.out.println(b.toString()); // visualizza 23:59
b.avanza(65);                  // ora b è l'una e quattro minuti
System.out.println(b.ore());   // visualizza 1
System.out.println(b.minuti()); // visualizza 4
System.out.println(b.toString()); // visualizza 01:04
```

Esercizio A4 (40%)

Scegliere e risolvere uno ed uno solo tra i seguenti esercizi A4-1 e A4-2.

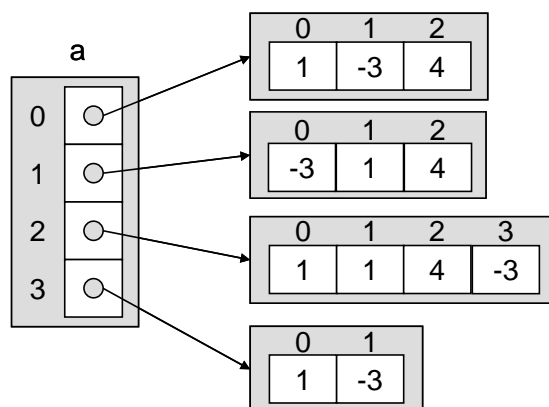
Indicare chiaramente sul foglio protocollo quale dei due esercizi si è deciso di risolvere.

Esercizio A4-1

Scrivere un metodo **boolean ripetuto(int[][] a, int val)** che:

- il parametro **a** è un array di array **a** non nullo in cui ciascun elemento è un array non nullo;
- verifica se ogni riga di **a** contiene almeno un elemento uguale a **val**.

Ad esempio, se **a** è il seguente array di array:



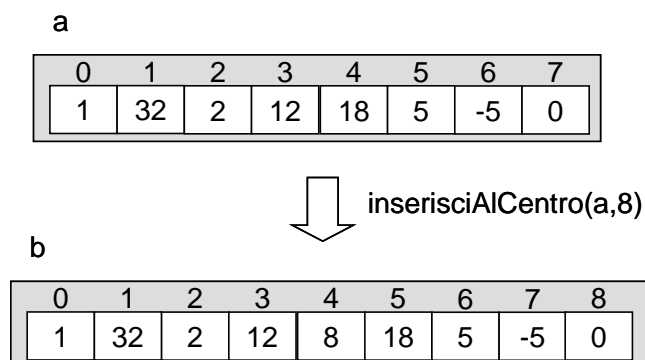
allora: **ripetuto(a,-3)** vale **true** (il -3 è ripetuto su tutte le righe, una volta per riga), **ripetuto(a,4)** vale **false** (il 4 non è ripetuto sull'ultima riga) e **ripetuto(a,1)** vale **true** (il 1 è ripetuto su tutte le righe).

Esercizio A4-2

Scrivere un metodo **int[] inserisciAlCentro(int[] a, int v)** che:

- ha come parametri un array **a** non nullo di interi e un valore intero **v**;
- crea e restituisce un nuovo array di interi **b** ottenuto da **a** inserendo, nella posizione centrale, un nuovo elemento uguale a **v**.

Ad esempio:



Si discuta esplicitamente il modo in cui si vuole gestire il caso in cui l'array **a** abbia lunghezza dispari anziché pari.