

---

**Esercizio 1**

Scrivere un metodo di classe **Intersezione** che presi in input due Array di **int** A e B, restituisce in output un array contenente gli elementi presenti sia in A che in B.

---

**Esercizio 2**

Scrivere un metodo di classe **identità** che prende in input un intero **n** e restituisce la matrice identità (che deve essere creata dal metodo) di ordine n (una matrice  $n \times n$  in cui tutti gli elementi sulla diagonale contengono 1 e tutti gli altri elementi 0).

n deve essere maggiore di 0, altrimenti il metodo restituisce **null**.

---

**Esercizio 3**

Scrivere un metodo di classe **trasposta** che prende in input una matrice A di numeri interi e restituisce una nuova matrice (che deve essere creata dal metodo) che sia la trasposta di A. (una matrice trasposta ha l'elemento (i,j) uguale a quello (j,i) della matrice di partenza).

---

**Esercizio 4**

Scrivere un metodo di classe **differenza** che prende in input due matrici A e B di numeri interi, e restituisce in output la matrice A-B (che deve essere creata dal metodo). Il metodo deve controllare che le matrici A e B siano rettangolari ed abbiano le stesse dimensioni, altrimenti solleva un'opportuna eccezione.

---

**Esercizio 5**

Scrivere un metodo di classe **reverse\_search** che prende in input due array A e B di int, e restituisce l'intero x se l'array B ha lunghezza minore o uguale dell'array A ed è possibile individuare un sottoarray di A di elementi di posizione consecutiva, a partire dalla posizione x di A, uguale all'array B letto al contrario, -1 altrimenti.

Per esempio, se  $A = \{70, 1, 55, 83, 21, 13, 43, 10, 56\}$ ; e  $B = \{43, 13, 21\}$ ; deve essere ritornato **4**.

---

**Esercizio 6**

Scrivere un metodo di classe **Unione** che presi in input due Array A e B del tipo generico T, restituisce in output un array contenente l'unione degli elementi contenuti in A e in B, con l'accortezza che gli elementi non devono essere ripetuti nell'array di output.

---

**Esercizio 7**

Scrivere un metodo di classe **coprimi** che prende in input due interi a e b, e restituisce in output **true** se e solo se a e b non hanno divisori interi maggiori di 1 in comune.

---

**Esercizio 8**

Scrivere un metodo di classe **somma** che prende in input due matrici A e B di numeri interi, e restituisce in output la matrice A+B (che deve essere creata dal metodo). Il metodo deve controllare che le matrici A e B siano rettangolari ed abbiano le stesse dimensioni, altrimenti solleva un'opportuna eccezione.

---

### Esercizio 9

Scrivere un metodo di classe **search** che prende in input due array A e B di int, e restituisce **true** se e solo se l'array B ha lunghezza minore o uguale dell'array A, ed è possibile individuare un sottoarray (di elementi consecutivi) di A uguale all'array B.

Per esempio, se  $A = \{1, 55, 83, 21, 13, 43, 10, 56\}$ ; e  $B = \{21, 13, 43\}$ ; deve essere ritornato **true**.

---

### Esercizio 10

Un array si dice **fortemente crescente** se ogni suo elemento (dal secondo all'ultimo) è almeno pari alla somma di tutti gli elementi che lo precedono.

Scrivere un metodo **public static boolean fc (int [] A)** che preso in input un array A di numeri interi, restituisce **true** se e solo se A è "fortemente crescente"

Ad esempio se  $A = [1, 1, 3, 5, 15]$  deve restituire **true**, mentre se  $A = [1, 2, 3, 5, 6]$  deve restituire **false**.

---

### Esercizio 11

Scrivere un metodo **public static boolean monotono (int [] A)** che preso in input un array A di numeri interi, restituisce **true** se e solo se A verifica le seguenti proprietà:

- Ha lunghezza pari
- La sua prima metà è ordinata in modo **strettamente crescente**
- La sua seconda metà è ordinata in modo **strettamente decrescente**.

Ad esempio se  $A = [1, 2, 20, 5]$  deve restituire **true**,  
mentre se  $A = [1, 2, 1, 5, 6]$  o se  $A = [1, 2, 2, 10]$  deve restituire **false**.

---

### Esercizio 12

Scrivere un metodo **public static ArrayList<Integer> estrai (int [] A, int k)** che preso in input un array di numeri interi ed un intero k, crea e restituisce un ArrayList di Integer in cui sono presenti tutti e soli gli interi presenti in A esattamente k volte, nello stesso ordine della loro prima occorrenza in A.

Ad esempio, se  $A = [3, 1, 5, 3, 9, 1]$  e  $k = 2$ , il metodo deve restituire un arrayList contenente gli Integer 3 ed 1.

---

### Esercizio 13

Scrivere un metodo **public static boolean monotono (int [] A)** che preso in input un array A di numeri interi, restituisce **true** se e solo se A è ordinato in modo non decrescente oppure non crescente.

Ad esempio se  $A = [1, 2, 2, 4, 5]$  o se  $A = [5, 4, 3, 3, 2]$  deve restituire true,  
mentre se  $A = [1, 2, 1, 5, 6]$  deve restituire false.

---

### Esercizio 14

Scrivere un metodo **public static boolean verifica (int [][] A, int [][] B, int [][] C)** che presi in input tre array bidimensionali di numeri interi restituisce **true** se e solo se valgono tutte le seguenti condizioni:

- A, B e C sono tutte e tre matrici rettangolari
- A, B e C hanno le stesse dimensioni
- $A + B = C$

---

### Esercizio 15

Scrivere un metodo **static int[] appendi (int[] a, int[] b)** che presi come parametri due array **a** e **b** di numeri interi restituisce un nuovo array avente lunghezza pari alla somma delle lunghezze degli array **a** e **b** e contenente prima tutti gli elementi di **a** e successivamente tutti gli elementi di **b**.

Ad esempio, se **a**=[0, 1, 2, 1] e **b**=[0, 20, 10, 1, 5] il metodo deve restituire l'array **[0, 1, 2, 1, 0, 20, 10, 1, 5]**.

---

### Esercizio 16

Scrivere un metodo **static void stampaTavolaPitagorica (int n, int m)** che preso come parametro due interi **n** ed **m** stampa a video una tavola pitagorica di dimensione **n** righe per **m** colonne.

Se ad esempio **n=4** e **m=5**, il metodo deve stampare

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
```

**Esercizio facoltativo:** fare in modo che i numeri risultino allineati nella stampa

---

### Esercizio 17

Scrivere un metodo **static int verificaTotocalcio (int[] colonnaVincente, int[] colonnaGiocata)** che presi come parametri due array **colonnaVincente** e **colonnaGiocata** di numeri interi avente entrambi lunghezza **n** e che rappresentano rispettivamente la colonna vincente e la colonna giocata di un concorso simile al totocalcio (in cui l'array contiene in posizione **i**, con **i** che varia da **0** ad **n-1**, il risultato della partita **i-esima**, che può essere **1** in caso di vincita della prima squadra, **2** in caso di vincita della seconda squadra e **0** in caso di pareggio) restituisce il numero di risultati esatti presenti in **colonnaGiocata** rispetto a **colonnaVincente**. Qualora le lunghezze dei due array siano diverse, oppure un qualche elemento contenuto nei due array non appartenga all'insieme **{0,1,2}**, viene restituito **-1**.

Ad esempio, se **colonnaVincente**=[0, 1, 2, 1] e **colonnaGiocata**=[0, 0, 0, 1] il metodo deve restituire **2** poiché vale sia che **colonnaVincente[0]=colonnaGiocata[0]** sia che **colonnaVincente[3]=colonnaGiocata[3]**.

---

### Esercizio 18

Scrivere un metodo **static int[] posizioniUguali (int[] a, int[] b)** che presi come parametri due array **a** e **b** di numeri interi avente entrambi lunghezza **n**, restituisce un array contenente tutte e sole le posizioni i per le quali vale che **A[i]=B[i]**. Nel caso in cui nessuna posizione **i** sia tale che **A[i]=B[i]**, deve essere restituito un array di lunghezza 0.

Ad esempio, se **A**=[11, 8, 14, 23] e **B**=[2, 8, 5, 23] il metodo deve restituire l'array [1, 3] poiché vale sia che **A[1]=B[1]** sia che **A[3]=B[3]**.

---

### Esercizio 19

Scrivere un metodo **static void stampaArray2Dim (int[][] a)** che preso come parametro un array bidimensionale **a** di numeri interi stampa a video una sua rappresentazione testuale in modo che:

- Ogni elemento dell'array **a** (che è a sua volta un array di interi) sia stampato su una singola linea, nell'ordine in cui tali array sono memorizzati nell'array bidimensionale **a**. Si noti come i vari array che compongono l'array **a** non devono necessariamente avere la stessa lunghezza.
- Ogni linea contenga gli elementi del corrispondente array di interi nello stesso ordine in cui sono memorizzati nell'array in questione e separati da un punto e virgola. Si noti che il punto e virgola non deve essere stampato dopo l'ultimo elemento di ogni linea.

---

### Esercizio 20

Scrivere un metodo **static int[] confronta (int[] a, int[] b)** che presi come parametri due array **a** e **b** di numeri interi di **uguale lunghezza** (qualora la lunghezza degli array **a** e **b** sia differente, deve essere restituito **null**) restituisce un nuovo array **c** di interi, avente la medesima lunghezza degli array **a** e **b**, e tale che, per ogni  $i=0, \dots, a.length-1$ ,

- **c[i] = 1** se  $a[i] > b[i]$
- **c[i] = 0** se  $a[i] = b[i]$
- **c[i] = -1** se  $a[i] < b[i]$

Ad esempio, se **a**=[3, 1, 20, 1] e **b**=[3, 20, 10, 1] il metodo deve restituire l'array **[0, -1, 1, 0]**.

---

### Esercizio 21

Scrivere un metodo **static void stampaCornice (int n, int m)** che preso come parametro due interi **n** ed **m** stampa a video una cornice di "\*" di dimensione orizzontale **n** e verticale **m**.

Se ad esempio **n=5** e **m=4**, il metodo deve stampare

```
*****
*      *
*      *
*      *
*****
```

---

### Esercizio 22

Dati due array **a** e **b** di numeri interi aventi lo stesso numero di elementi **n**, si dice che **a domina b** se e solo se per tutte le posizioni possibili **i** dell'array **a** (da **0** a **n-1**) **a[i]** è almeno pari alla somma di tutti gli elementi presenti nell'array **b** dalla posizione **i** fino all'ultima posizione.

Ad esempio, l'array [11, 8, 14] domina l'array [2, 5, 3] in quanto  $11 \geq 2+5+3$ ,  $8 \geq 5+3$  e  $14 \geq 3$ .

Scrivere un metodo **static boolean domina (int[] a, int[] b)** che presi come parametro due array **a** e **b** di numeri interi restituisce **true** se e solo se gli array **a** e **b** hanno lo stesso numero di elementi ed **a** domina **b**.

---

### Esercizio 23

Scrivere un metodo **static int[] soloPari (int[] a)** che preso come parametro un array **a** di numeri interi restituisce un nuovo array avente come elementi, nello stesso ordine di **a**, tutti i **numeri pari** presenti in **a**. La dimensione dell'array restituito deve essere pertanto uguale al numero di interi pari presenti in **a**.

Ad esempio, se **a** è [10,3,20,4,6,5,1], il metodo deve restituire l'array [10,20,4,6].

---

#### Esercizio 24

Dato un array, il suo inverso è dato dall'array in cui la prima cella diviene l'ultima, la seconda la penultima e così via (l'ultima cella diviene la prima).

Dati due array **a** e **b** di numeri interi aventi lo stesso numero di elementi **n**, il prodotto scalare dei due array è dato dalla somma dei prodotti degli elementi corrispondenti.

Ad esempio, il prodotto scalare tra [11, 8, 14] e [2, 5, 1] è  $11*2 + 8*5 + 14*1 = 76$ .

Scrivere un metodo **static int prodottoScalare (int[] a, int[] b)** che presi come parametro due array **a** e **b** di numeri interi restituisce il prodotto scalare di **a** per l'inverso di **b**. Il metodo **non deve** modificare gli array **a** e **b**.

---

#### Esercizio 25

Scrivere un metodo **static int[][] allocaMatrice (int[] a)** che preso come parametro un array **a** di numeri interi restituisce un nuovo array bidimensionale avente numero di righe uguale alla lunghezza dell'array **a**, e tale che, per ogni  $i=0, \dots, a.length-1$ , la lunghezza della riga di indice **i** sia uguale **i** sia uguale al contenuto della *i*-esima posizione dell'array **a**. Il metodo **non deve** modificare l'array **a**.

---

#### Esercizio 26

Scrivere un metodo **static int[] sottoMedia (int[] a)** che preso come parametro un array **a** di numeri interi restituisce un nuovo array formato da tutti e soli gli interi di **a** che siano non maggiori della media di tutti gli elementi contenuti nell'array **a**.

---

#### Esercizio 27

Scrivere senza far uso di costrutti iterativi un metodo **static int max (int[] a)** che preso come parametro un array **a** di numeri interi restituisce il massimo intero presente nell'array. Si possono usare metodi ausiliari, ma sempre senza utilizzare costrutti iterativi al loro interno.

---

#### Esercizio 28

Scrivere un metodo iterativo

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

che, presi come parametro due array **a** e **b** di numeri interi aventi la stessa lunghezza, restituisce un array di lunghezza pari alla somma delle lunghezze di **a** e **b**, in cui gli elementi di **a** sono alternati a quelli di **b**, ed è preservato l'ordine in cui gli elementi compaiono nei singoli array.

Se **a** e **b** non hanno la stessa lunghezza, viene restituito **null**.

Ad esempio, se  $a=[1, 16, 10, 4]$  e  $b=[6, 2, 1, 70]$  viene restituito  $[1, 6, 16, 2, 10, 1, 4, 70]$ .

---

#### Esercizio 29

Scrivere un metodo iterativo

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

che, presi come parametro due array **a** e **b** di numeri interi aventi la stessa lunghezza, restituisce un array della stessa lunghezza degli array **a** e **b**, in cui in ogni posizione **i** ( $i=0, \dots, a.length-1$ ) è presente il massimo tra gli elementi presenti in **a** e **b** in posizione **i**.

Se **a** e **b** non hanno la stessa lunghezza, viene restituito **null**.

Ad esempio, se  $a=[1, 16, 10, 4]$  e  $b=[6, 2, 1, 70]$  viene restituito  $[6, 16, 10, 70]$ .

---

### Esercizio 30

Scrivere un metodo iterativo

```
public static int distinti (int[] a)
```

che, preso come parametro un array di numeri interi, restituisce il numero degli elementi distinti presenti nell'array.

Ad esempio, se  $a=[1, 16, 10, 4, 16, 2, 1, 16]$  viene restituito 5.

---

### Esercizio 31

Un array si dice **equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre equilibrati.

Ad esempio, l'array  $[1, 1, 10, 4, 6, 2]$  è equilibrato in quanto  $1+1+10 = 4+6+2$ .

Scrivere un metodo iterativo

```
public static boolean equilibrato (int[] a)
```

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è equilibrato.

---

### Esercizio 32

Un array si dice **totalmente equilibrato** se la somma degli elementi nella sua prima metà è uguale alla somma degli elementi nella sua seconda metà, ed inoltre anche il sottoarray costituito dalla sua prima metà e quello costituito dalla sua seconda metà sono totalmente equilibrati (qualora l'array abbia lunghezza dispari, si assuma che l'elemento centrale non appartiene né alla prima né alla seconda metà). Un array vuoto o un array di un solo elemento sono sempre totalmente equilibrati.

Ad esempio, l'array  $[1, 3, 1, 2, 1, 2]$  è totalmente equilibrato in quanto  $1+3+1 = 2+1+2$ , ed inoltre anche i sottoarray  $[1, 3, 1]$  e  $[2, 1, 2]$  sono totalmente equilibrati poiché  $1=1$  e  $2=2$ , rispettivamente; infine, i sottoarray  $[1]$ ,  $[1]$ ,  $[2]$  e  $[2]$  sono banalmente totalmente equilibrati poiché composti da un solo elemento.

Scrivere un metodo ricorsivo

```
public static boolean totalmenteEquilibrato (int[] a)
```

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è totalmente equilibrato.

---

### Esercizio 33

Scrivere un metodo iterativo

```
public static int[] filtro (int[] origine, int[] filtro)
```

che, presi come parametri due array di numeri interi, restituisce un nuovo array di numeri interi di lunghezza uguale a quella dell'array **filtro** e tale che la prima posizione contenga l'elemento in posizione **filtro[0]** dell'array **origine**, la seconda posizione l'elemento in posizione **filtro[1]** e così via. Se l'array **filtro** contiene valori negativi o maggiori della o uguali alla lunghezza dell'array **origine**, viene restituito **null**.

Ad esempio, l'array **origine** è  $[1, 16, 10, 4, 6, 2]$  e l'array **filtro** è  $[4,0,1]$  viene restituito l'array  $[6,1,16]$ .

---

### Esercizio 34

Scrivere un metodo iterativo

```
public static int piuFrequente (int[] a)
```

che, preso come parametro un array di numeri interi, restituisce uno degli elementi dell'array aventi frequenza massima (la frequenza è il numero di occorrenze dell'elemento nell'array).

Ad esempio, se  $a=[1, 16, 10, 4, 16, 2]$  viene restituito 16.

---

### Esercizio 35

Un numero intero è un quadrato perfetto se può essere espresso come il quadrato di un altro numero intero. Ad esempio, **81** è un quadrato perfetto in quanto è il quadrato di 9.

Scrivere un metodo

**public static boolean quadratoPerfetto (int n)**

che, preso come parametro un numero intero, restituisce **true** se e solo se **n** è un quadrato perfetto.

---

### Esercizio 36

Scrivere un metodo

**public static boolean ordinato (int[] a)**

che, preso come parametro un array di numeri interi, restituisce **true** se e solo se **a** è ordinato in modo **non decrescente**.

---

### Esercizio 37

Scrivere un metodo

**public static int sommaMassima (int[] a)**

che preso come parametro un array di numeri interi, restituisce la somma del sottoarray (di elementi consecutivi) avente somma massima in **a**.

Ad esempio, se **a = {-10, 35, -20, 50, -30, -14, 2, 19}** il metodo deve restituire **65** in quanto il sottoarray {35, -20, 50} è quello di somma massima.

---

### Esercizio 38

Risolvere l'esercizio **37** in modo ricorsivo, senza far uso di nessun comando iterativo. Si sfrutti la tecnica del divide et impera.

---

### Esercizio 39

Scrivere un metodo

**static int contaSopraSoglia (int[] a, int n)**

che, preso come parametro un array di numeri interi **a** ed un intero **n**, restituisce il numero di elementi di **a** che sono maggiori o uguali ad **n**.

Scrivere quindi un metodo

**static int[] elencoSopraSoglia (int[] a, int n)**

che, preso come parametro un array di numeri interi **a** ed un intero **n**, restituisce un nuovo array **della opportuna lunghezza** contenente tutti e soli gli elementi di **a** che sono maggiori o uguali ad **n**.

---

### Esercizio 40

Dato un array **a** di numeri interi, un suo elemento **n** si dice **veritiero** se compare nell'array **a** esattamente **n** volte.

Scrivere un metodo

**public static int contaVeritieri (int[] a)**

che preso come parametro un array di numeri interi, restituisce il numero di interi distinti **veritieri** presenti in **a**.

Ad esempio, se **a = {-10, 4, 2, 2, 3, 4, 1, 4, 4, 3}** il metodo deve restituire **3** in quanto l'array contiene 3 elementi veritieri: 1, 2 e 4.

---

#### Esercizio 41

Scrivere un metodo iterativo

**public static int[] scostamentoDaMedia (int[] a)**

che, preso come parametro un array di numeri interi, restituisce un array di numeri interi della stessa lunghezza di **a**, in cui in posizione  $i$  ( $i=0,\dots,a.length$ ) è inserito il valore  $a[i]-M$ , dove  $M$  rappresenta la media aritmetica dei valori contenuti in **a**.

---

#### Esercizio 42

Scrivere un metodo iterativo

**public static boolean permutati (int[] a, int[] b)**

che, presi come parametri due array di numeri interi, **senza modificare il contenuto degli array**, restituisce **true** se e solo se **a** e **b** possiedono gli stessi elementi (tenendo conto anche degli elementi ripetuti), non necessariamente nello stesso ordine.

Ad esempio, se  $a=\{1,3,5,3,7\}$  e  $b=\{7,3,3,1,5\}$  il metodo deve restituire **true**.

---

#### Esercizio 43

Scrivere un metodo iterativo

**public static int[] estraiRipetuti (int[] a, int k)**

che, preso come parametro un array **a** di numeri interi ed un intero **k**, restituisce un array di numeri interi (della opportuna lunghezza) contenente tutti e soli gli elementi che si ripetono in **a** esattamente **k** volte.

Se nessun elemento di **a** è presente **k** volte, viene restituito un array di lunghezza 0.

Se **a** vale null, viene restituito null.

Ad esempio, se  $a=\{1, 4, 3, 1, 5, 6, 3, 2, 8, 4, 4\}$  e  $k=2$ , viene restituito l'array  $\{1, 3\}$  in quanto sia l'intero 1 che l'intero 3 sono presenti 2 volte in **a**.

---

#### Esercizio 44

Scrivere un metodo iterativo

**static int massimoComune (int[] a, int[] b)**

che, presi come parametri due array **a** e **b** di numeri interi, restituisce il massimo valore presente in entrambi gli array. Se nessun elemento di **a** è presente in entrambi gli array, o (almeno) un array è vuoto o inesistente, viene restituito -1.

Ad esempio, se  $a=\{1, 4, 3, 1, 5, 6, 3, 2, 8, 42, 4\}$  e  $b=\{11, 40, 30, 1, 5, 6, 3, 2, 8\}$ , viene restituito 8.

---

#### Esercizio 45

Scrivere un metodo iterativo

**static int massimoCondizionato (int[] a, int x)**

che, presi come parametri un array **a** di numeri interi ed un intero **x**, restituisce il massimo valore presente nell'array **a** che compare nell'array almeno **x** volte. Se nessun elemento di **a** è occorre almeno **x** volte, viene restituito il massimo valore presente nell'array. Si può assumere che l'array sia esistente e non vuoto.

Ad esempio, se  $a=\{11, 4, -1, 4\}$  e  $x=2$ , viene restituito 4.

---

#### Esercizio 46

In un array **a**, un **punto fisso** è un elemento uguale alla posizione, cioè l'elemento di posizione **i** è un punto fisso se  $a[i]=i$ .

Ad esempio, se  $a=\{11, -4, 2, 4\}$ , l'elemento in terza posizione (posizione 2) è un punto fisso.

1. Scrivere un metodo

**static int contaPuntiFissi (int[] a)**

che, preso come parametro un array **a** di numeri interi, restituisce il numero di punti fissi presenti in **a**.

2. Scrivere un metodo

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

che, preso come parametro un array **a** di numeri interi, restituisce un nuovo array contenente tutti e soli gli elementi che sono punti fissi in **a**. Se in **a** non sono presenti punti fissi, viene restituito un array di lunghezza 0.

3. Scrivere un metodo

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

che, preso come parametro un array **a** di numeri interi, restituisce il massimo punto fisso presente in **a**. Se in **a** non sono presenti punti fissi, viene restituito **-1**.

---

#### Esercizio 47

Scrivere un metodo

```
static int[] massimo (int[] a, int[] b)
```

che, presi come parametro due array **a** e **b** di numeri interi, restituisce un nuovo array di numeri interi avente

1. lunghezza uguale al massimo valore tra la lunghezza di **a** e quella di **b**
2. In posizione **i** il massimo tra i valori **a[i]** e **b[i]**, se entrambi sono definiti; altrimenti, l'unico valore definito (tra **a[i]** e **b[i]**).

Ad esempio, se  $a=\{1,5,8,3\}$  e  $b=\{3,4\}$ , il metodo deve restituire l'array  $\{3,5,8,3\}$ .