

# Unique

---

Scrivere un metodo, chiamato `unique`, che dato in input un array `a` di numeri interi positivi (eventualmente vuoto), restituisca l'array (eventualmente vuoto) contenente gli elementi di `a` che appaiono una sola volta in `a` stesso, nello stesso ordine con cui essi appaiono in `a`.

ESEMPI

Input: {32,23,12,23,15,32}

Output: {12,15}

Input: {127,34,45,56,34,127,23,45,12,56,23}

Output: {12}

Input: {20,11,15,41,23}

Output: {20,11,15,41,23}

Input: {127,34,45,56,34,127,23,45,12,56,23,12}

Output: {}

Input: {}

Output: {}

# Frequencies

---

Scrivere un metodo, detto `frequencies`, che, dato in input un array `a` di numeri interi non vuoto, restituisca l'array delle frequenze degli elementi distinti di `a`, in ordine crescente dei valori degli elementi. Ad esempio, con input il vettore `{1,6,5,8,6,6,8,5,3,1,6,8}`, il metodo deve restituire l'array `{2,1,2,4,3}`, in quanto il valore 1 appare due volte, il 3 una volta, il 5 due volte, il 6 quattro volte e l'8 tre volte. L'array in input non deve essere modificato dall'esecuzione del metodo.

## ESEMPI

Input: `{1,6,5,8,6,6,8,5,3,1,6,8}`

Output: `{2,1,2,4,3}`

Input: `{6,6,6,6}`

Output: `{4}`

Input: `{3,5,6,2}`

Output: `{1,1,1,1}`

# Distribuisci

---

Un array di numeri interi positivi si dice  $d$ -uniforme, per qualche  $d > 0$ , se è ordinato in modo non crescente e se ogni valore appare esattamente  $d$  volte. Ad esempio, l'array  $\{13,13,13,8,8,8,6,6,6\}$  è 3-uniforme. La distribuzione di un array  $d$ -uniforme contenente  $n$  elementi, è l'array ottenuto ponendo le  $d$  copie di ciascun valore in  $n/d$  gruppi ordinati in modo decrescente, uno di seguito all'altro. Ad esempio la distribuzione dell'array  $\{13,13,13,8,8,8,6,6,6\}$  è l'array  $\{13,8,6,13,8,6,13,8,6\}$ .

Scrivere un metodo in Java--, detto **distribuisci**, che, dato in input un array  $a$  di numeri interi positivi ordinato in modo non crescente, restituisca la distribuzione di  $a$ , se  $a$  è  $d$ -uniforme per qualche  $d > 0$ , altrimenti restituisca l'array  $a$  stesso.

Ad esempio, con input l'array  $\{13,13,13,8,8,8,6,6,6\}$  che è 3-uniforme, il metodo deve restituire l'array  $\{13,8,6,13,8,6,13,8,6\}$ , mentre con input l'array  $\{13,13,13,8,8,8,6,6,6\}$  che non è  $d$ -uniforme per alcun  $d > 0$ , il metodo deve restituire l'array  $\{13,13,13,8,8,8,6,6,6\}$ . Infine, con input l'array  $\{13,13,13,13\}$  che include un solo valore ed è 4-uniforme, il metodo deve restituire l'array  $\{13,13,13,13\}$ .

# Tic-Tac-Toe

## Esercizio

Una matrice quadrata binaria  $a$  di dimensione  $n$  contenente solo 0, 1 e 2 si dice essere una *configurazione finale* se esiste una riga, una colonna o una diagonale contenente  $n$  simboli 1 oppure se esiste una riga, una colonna o una diagonale contenente  $n$  simboli 2.

Scrivere un metodo chiamato `tictactoe` che, data in input una matrice quadrata binaria  $a$  tale che  $a$  contenga solo 0, 1 e 2, restituisca il valore `true` se e solo se la matrice è una configurazione finale.

## Esempi

- $a = \{\{2, 1, 0\}, \{0, 2, 1\}, \{0, 0, 2\}\}$ : `true`
- $a = \{\{2, 2, 1\}, \{0, 1, 1\}, \{1, 0, 2\}\}$ : `true`
- $a = \{\{2, 1, 2\}, \{1, 1, 1\}, \{2, 0, 0\}\}$ : `true`
- $a = \{\{2, 1, 2\}, \{2, 1, 1\}, \{2, 0, 0\}\}$ : `true`
- $a = \{\{2, 1, 2\}, \{2, 1, 1\}, \{1, 2, 1\}\}$ : `false`
- $a = \{\{0, 1, 2\}, \{2, 0, 1\}, \{1, 2, 0\}\}$ : `false`

2	1	0
0	2	1
0	0	2

2	2	1
0	1	1
1	0	2

2	1	2
1	1	1
2	0	0

2	1	2
2	1	1
2	0	0

Quattro configurazioni finali

2	1	2
2	1	1
1	2	1

0	1	2
2	0	1
1	2	0

Due configurazioni non finali