

# Code kata

Vettori, liste, matrici

## **Indice generale**

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>Generare una sequenza di interi progressivi.....</b>          | <b>3</b>  |
| <b>2</b>  | <b>Modificare gli elementi di un vettore.....</b>                | <b>4</b>  |
| <b>3</b>  | <b>Ricerca di un elemento in un vettore.....</b>                 | <b>5</b>  |
| <b>4</b>  | <b>Invertire la posizione degli elementi.....</b>                | <b>6</b>  |
| <b>5</b>  | <b>Generare una sequenza di numeri interi casuali.....</b>       | <b>7</b>  |
| <b>6</b>  | <b>Generare una sequenza di interi casuali non ripetuti.....</b> | <b>8</b>  |
| <b>7</b>  | <b>Filtrare un vettore e restituire una lista.....</b>           | <b>9</b>  |
| <b>8</b>  | <b>Caricare i dati da una lista a un vettore.....</b>            | <b>10</b> |
| <b>9</b>  | <b>Calcolare i totali di riga di una matrice.....</b>            | <b>11</b> |
| <b>10</b> | <b>Calcolare i totali di colonna di una matrice.....</b>         | <b>12</b> |
| <b>11</b> | <b>Calcolare la funzione di 2° grado.....</b>                    | <b>13</b> |

# 1 Generare una sequenza di interi progressivi

---

## Problema

Generare una sequenza di numeri interi progressivi che cadano in un intervallo. Restituire la sequenza in un vettore. Se l'estremo sinistro e destro dell'intervallo sono uguali deve essere restituito un vettore vuoto (di lunghezza zero).

Esempio:

```
int da = 5; // estremo sinistro intervallo
int a = 10; // estremo destro intervallo: a >= da
//-> 5,6,7,8,9,10
```

## Variazioni

Fornire delle versioni con ciclo `for`, `while`, `do while`.

## 2 Modificare gli elementi di un vettore

---

### Problema

Dato un vettore di `double`, aumentare tutti gli elementi del 10%.

Esempio:

```
double[] dati = { 1000, 1200, 1500 };  
// -> 1100, 1320, 1650
```

### Variazioni

1. Generalizza il procedimento per l'aumento di un valore percentuale qualsiasi.

```
double[] dati = { 1000, 1200, 1500 };  
double perc = 20; // 20%  
// -> 1200, 1440, 1800
```

2. Fornisci delle versioni con ciclo `for`, `while`.
3. Modifica il procedimento, in modo che venga restituito un nuovo vettore e non venga modificato il vettore originale.

## 3 Ricerca di un elemento in un vettore

---

### Problema

Dato un vettore di interi e dato un numero, restituire la posizione del numero nel vettore. Restituire -1 se il numero non è presente nel vettore.

Esempio:

```
int[] dati = { 10, 30, 45, -30 };  
int numero = 30;  
// -> posizione: 1  
  
int numero2 = 75;  
// -> posizione: -1
```

### Variazioni

1. Fornisci versioni del procedimento con il ciclo `while`, `for` e `foreach`.
2. Modifica il procedimento in modo che venga restituito `true` se esiste, `false` se non esiste.

```
int[] dati = { 10, 30, 45, -30 };  
int numero = 30;    // -> true  
  
int numero2 = 75;   // -> false
```

## 4 Invertire la posizione degli elementi

---

### Problema

Dato un vettore, invertire la posizione dei suoi elementi. (Modificare il vettore)

Esempio:

```
int[] dati = { 10, 30, 45, -30 };  
// -> -30, 45, 30, 10
```

### Variazioni

1. Fornisci versioni del procedimento con i cicli `while` e `for`.
2. Modifica il procedimento, in modo che venga restituito un nuovo vettore contenente gli stessi elementi dell'originale ma in posizione invertita.

## 5 Generare una sequenza di numeri interi casuali

### Problema

Generare una sequenza di interi casuali; ogni numero casuale deve cadere in un intervallo specificato. La sequenza deve essere restituita in un vettore. Se la lunghezza della sequenza è zero deve essere restituito un vettore vuoto.

Esempio:

```
int da = 1; // estremo sinistro intervallo numero casuale
int a = 20; // estremo destro intervallo numero casuale: a > da
int n = 10; // lunghezza sequenza n >= 0
//-> 5, 1, 18, 12, 10, 5, 8, 19, 12, 8
```

### Variazioni

Fornisci versioni con ciclo `for`, `while` e `do while`.

### Requisiti (necessari / utili) per la soluzione

Generare un numero casuale; oggetto `Random`, metodo `Next()`:

```
Random rnd = new Random();
int casuale = rnd.Next(1, 10); //-> casuale 1 - 9 (compresi)
```

## 6 Generare una sequenza di interi casuali non ripetuti

### Problema

Generare una sequenza di interi casuali; ogni numero casuale deve cadere in un intervallo specificato. La sequenza non deve contenere elementi ripetuti. Se la lunghezza della sequenza è zero deve essere restituito un vettore vuoto.

Esempio:

```
int da = 1; // estremo sinistro intervallo numero casuale
int a = 20; // estremo destro intervallo numero casuale: a > da
int n = 10; // lunghezza sequenza: n <= (a - da + 1)
//-> 5, 1, 18, 12, 10, 7, 8, 19, 13, 9
```



## 7 Filtrare un vettore e restituire una lista

---

Dato un vettore di interi, creare una nuova lista contenente soltanto i valori negativi.

Esempio:

```
int[] dati = { -10, 30, 45, -30 };  
// -> -10, -30
```

### ***Variazioni***

Fornisci versioni con i cicli `foreach` e `for`.

## 8 Popolare un vettore con i dati di una lista

---

Data una lista di stringhe, creare un nuovo vettore che contenga una copia dei dati.

Esempio:

```
List<string> dati = new List<string>() {"10", "20", "30", "40"},  
// -> "10", "20", "30", "40"
```

### ***Variazioni***

Fornisci versioni con i cicli `foreach` e `for`.

## 9 Calcolare i totali di riga di una matrice

### Problema

Sia data una matrice `double` Nx4 che contiene dei dati nelle prima 3 colonne. Calcolare il totale di ogni riga, memorizzandolo nell'ultima colonna

Esempio:

```
double[,] dati =
{
    {10, 20, 30, 0},
    {100, 200, 300, 0},
    {1000, 2000, 3000, 0},
    {10000, 20000, 30000, 0},
};
//->
//    {10, 20, 30, 60},
//    {100, 200, 300, 600},
//    {1000, 2000, 3000, 6000},
//    {10000, 20000, 30000, 60000},
```

### Variazioni

Supponi che la matrice contenga dati in tutte le colonne. Modifica il procedimento in modo che i totali di riga siano memorizzati in un vettore (la lunghezza del vettore deve corrispondere al numero di righe della matrice).

## 10 Calcolare i totali di colonna di una matrice

### Problema

Sia data una matrice `double` Nx4 che contiene dei dati nelle prima N-1 righe. Calcolare il totale di ogni colonna, memorizzandolo nell'ultima riga

Esempio:

```
double[,] dati =
{
    {10, 20, 30, 40},
    {100, 200, 300, 400},
    {1000, 2000, 3000, 4000},
    {0, 0, 0, 0},
};
//->
//    {10, 20, 30, 40},
//    {100, 200, 300, 400},
//    {1000, 2000, 3000, 4000},
//    {1110, 2220, 3330, 4440},
```

### Variazioni

Supponi che la matrice contenga dati in tutte le righe. Modifica il procedimento in modo che i totali di colonna siano memorizzati in un vettore (la lunghezza del vettore deve corrispondere al numero di colonne della matrice).

## 11 Calcolare la funzione di 2° grado

---

### Problema

Sia data una matrice `double` Nx2 contenente, nella prima colonna, le X della funzione di 2° grado:

$$y = 2x^2 + x$$

Calcolare le Y e memorizzarle nella seconda colonna.

```
double[,] dati =  
{  
    {1, 0},  
    {2, 0},  
    {3, 0},  
    {5, 0},  
};  
//->  
//    {1, 3},  
//    {2, 10},  
//    {3, 21},  
//    {5, 55},
```