

Cognome e Nome		Corso		Matricola	
----------------	--	-------	--	-----------	--

Esercizio 1

Si consideri il seguente metodo:

```
public class Esercizio1C {
    public static boolean metodoC(int x[]) {
        boolean s=false;
        for (int i=1;i<x.length;i++)
            if (!s && x[i-1]>x[i]) s=!s;
            else if (s && x[i-1]<x[i]) s=!s;
            else return false;
        return true;
    }
    public static void main(String[] args) {
        int [] v ={5,2,7,3,5,1,7};
        boolean b=metodoC(v);
        System.out.print(b);
    }
}
```

Si illustri il funzionamento del metodo *metodoC*. In particolare si mostri cosa viene stampato in output nel caso in cui l'array passato in input sia **V** = [5, 2, 7, 3, 5, 1, 7].

Esercizio 2

Si scriva un metodo *verificaVettore* che riceve in ingresso due vettori di interi **v** e **w**, e restituisce un valore booleano. In particolare restituisce **true** se per ogni elemento di **v** esiste in **w** un elemento con lo stesso valore assoluto ma di segno opposto, **false** altrimenti.

Ad esempio, nel caso in cui **v** = [20, -9, 4, -8], restituisce **true** se **w** = [1, 9, -20, 4, -4, 0, 8], **false** se **w** = [1, 9, 4, 0, 8].

Esercizio 3

Si realizzi una classe *Matrice* per rappresentare matrici di interi che contenga almeno i seguenti metodi:

1. un metodo *creaArray* che riceve in ingresso una matrice **M** ed un intero **p** e restituisce in output un array contenente tutti gli elementi di **M** strettamente maggiori di **p** presenti sulle colonne pari;
2. un metodo *estraiCroce* che riceve in ingresso una matrice quadrata **M** di dimensione dispari e restituisce in output una matrice contenente gli elementi di **M** presenti sulla riga e sulla colonna centrali escluso l'elemento centrale;
3. un metodo *verificaMatrice* che riceve in ingresso una matrice **M** e restituisce in output un booleano. In particolare il metodo restituisce **true** se ogni colonna di **M** non contiene elementi duplicati, **false** altrimenti;
4. un metodo *main* che legge una matrice quadrata **M** di numeri interi e dimensione dispari e provvede, invocando opportunamente i metodi proposti, a realizzare i compiti di cui ai punti (1), (2) e (3) .

Esempio:

$$M = \begin{bmatrix} 6 & 4 & 11 & 9 & 12 \\ 2 & 1 & 10 & 7 & 10 \\ 14 & 21 & 17 & 1 & 4 \\ 8 & 10 & 13 & 3 & 1 \\ 10 & 2 & 12 & 10 & 30 \end{bmatrix}$$

- Il metodo *creaArray* invocato sulla matrice **M** con **p**=10 restituisce l'array **V** = [14, 11, 17, 13, 12, 12, 30]
- Il metodo *estraiCroce* invocato sulla matrice **M**, restituisce la matrice

$$\begin{bmatrix} 14 & 11 \\ 21 & 10 \\ 1 & 13 \\ 4 & 12 \end{bmatrix}$$
- Il metodo *verificaMatrice*, invocato sulla matrice **M** restituisce **true**.