

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

TRACCIA C

Esercizio 1

Si consideri il seguente programma:

```
public class Esercizio1C {
    public static boolean metodo1C(int[] v){
        boolean mult=v.length>0;
        for (int i=1; i<=v.length && mult; i++)
            if (v[i-1]%10!=i)
                mult=false;
        return mult;
    }

    public static void main(String args[]) {
        int v[] = {11, 12, 20, 16, 8, 16};
        boolean s = metodo1C(v);
        System.out.println (s);
    }
}
```

Si descriva sinteticamente la funzione svolta dal metodo **metodo1C** e, in particolare, si mostri l'esecuzione e cosa viene stampato nel caso in esempio, in cui $v = [11, 12, 20, 16, 8, 16]$.

Esercizio 2

Si scriva un metodo *annullaSimmetrici* che riceve in ingresso un vettore di interi v , e restituisce un vettore di interi w ottenuto ponendo a zero le coppie di elementi simmetrici uguali.

Ad esempio, se $v = [7, 3, 6, 6, 1, 8, 8, 6, 3, 5]$, il vettore restituito è $w = [7, 0, 0, 6, 1, 8, 8, 0, 0, 5]$.

Esercizio 3

Si scriva una classe *Esercizio3C* che contenga i seguenti metodi:

1. Un metodo *righeDecrescenti* che riceve una matrice di interi M e restituisce *true* se e solo se ogni riga di M è a valori decrescenti. Per maggiore chiarezza si veda l'esempio.
2. Un metodo *controlla* che riceve in ingresso una matrice di interi M ed un intero k , e restituisce *true* se e solo se la sottomatrice di M ottenuta eliminando la riga di indice k contiene solo valori non nulli. Per maggiore chiarezza si veda l'esempio.
3. Un metodo *costruisci* che riceve in ingresso una matrice di interi M e restituisce una matrice R ottenuta ponendo l'elemento $R[i][j]$ pari alla somma degli elementi presenti sulla j -esima colonna di M . Per maggiore chiarezza si veda l'esempio.
4. Un metodo *main* nel quale si legge una matrice di interi, e si invocano opportunamente i metodi definiti ai punti 1, 2 e 3.

Esempio:

$M =$

12	9	8	5	3
5	0	-1	-3	-10
15	12	13	6	3
9	8	5	3	-2
9	7	5	3	1

1. *righeDecrescenti* (M) restituisce *true*

2. *controlla* (M , 1) restituisce *true*

3. *costruisci* (M) restituisce la matrice $R =$

50	36	30	14	-5
50	36	30	14	-5
50	36	30	14	-5
50	36	30	14	-5
50	36	30	14	-5