

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

Esercizio 1

Si consideri il seguente programma:

```
public class Esercizio1D{
    public static int [] annulla(int[] a, int[] b)
    {
        int [] risultato = new int[a.length];
        if (a.length != b.length)
            return risultato;
        for (int i = 0; i < a.length; i++)
        {
            if (a[i] / b[i] == 0)
                risultato[i] = a[i];
            else if (b[i] / a[i] == 0)
                risultato[i] = b[i];
            else
                risultato[i] = 0;
        }
        return risultato;
    }

    public static void main(String args[]){
        int[] b = {5, 128, 56, 15, 6, 12, 3 };
        int[] a = {2, 2, 112, 15, 2, 3, 4 };
        int[] c = annulla(b, a);
    }
}
```

Si illustri il funzionamento del metodo *annulla* e si mostri il contenuto del vettore *c* al termine dell'esecuzione del *main*.

Esercizio 2

Scrivere un metodo *calcolaVettore* che riceve un vettore **A** di interi ed un intero **k** (minore della lunghezza di **A**) e restituisce un vettore di double i cui elementi siano dati dal prodotto tra l'elemento di **A** in posizione **i-esima** e la media dei primi **k** elementi **A** se **i** è maggiore di **k** e dal prodotto tra l'elemento di **A** in posizione **i-esima** e la media degli ultimi **k** elementi se **i** è minore o uguale a **k**. Ad es. se **A**=[3, 2, 3, 8, 9, 3] e **k** = 2 il metodo restituirà il vettore [18, 12, 18, 20, 22.5, 7.5].

Esercizio 3

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

1. un metodo *verificaMatrice* che riceve una matrice di interi **M**, un intero **x** ed un intero **y** e restituisce **true** se il rapporto tra la somma dei primi **x** elementi ed il prodotto degli ultimi **x** elementi di ciascuna colonna è maggiore di **y**;
2. un metodo *calcolaVettore* che riceve una matrice di interi **M** e restituisce un vettore costruito mettendo in riga tutte le colonne della matrice **M** in ordine inverso se l'indice di riga è pari, in ordine diretto se l'indice è dispari;
3. un metodo *calcola* che riceve una matrice di interi **M** di dimensione **m**x**n** ed un vettore **V** di dimensione 2 e restituisce la sottomatrice di **M** individuata nel vertice inferiore destro dalla posizione (**m**-1,**n**-1), e nel vertice superiore sinistro dagli elementi di **V**.
4. un metodo *main* che invochi opportunamente i metodi descritti nei punti precedenti.

Esempio
M=

0	0	1	3	2
0	1	6	1	0
1	4	3	1	1
2	2	0	0	2

- *verificaMatrice*(M, 2, 12) = false
- *calcolaVettore*(M) = [2 1 0 0 0 1 4 2 0 3 6 1 3 1 1 0 2 1 0 2]

- V=[2,3]
- *calcola*(M, V)=

1	1
0	2