

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

Esercizio 1

Si consideri il seguente programma:

```
public class Eserciziola{
    static void elabora(int[] v) {
        for (int i = 0; i < v.length; i++) {
            int s = 0;
            if (i % 2 == 0) {
                for (int j = i + 1; j < v.length; j++)
                    s += v[j];
            }else{
                for (int j = i - 1; j >= 0; j--)
                    s += v[j];
            }
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        int[] a = { 3, -2, 4, 7, 3, -1 };
        elabora(a);
    }
}
```

Si illustri il funzionamento del metodo *elabora* e, a titolo d'esempio, si mostri cosa viene stampato in output al termine dell'esecuzione del *main*.

Esercizio 2

Scrivere un metodo *verifica* che riceve un vettore **a** di **interi** (di dimensione pari) ed un intero **x**. Il metodo restituisce il valore **true** se per **ogni** elemento di **a** in posizione **pari** la sua somma con l'elemento successivo è minore o uguale a **x** ed il suo prodotto con l'elemento successivo è maggiore o uguale a **x**, **false** altrimenti. Ad esempio, se **a**=[7, 3, 3, 5, 8, 3, 9, 2] e **x**=15 il metodo restituirà **true**.

Esercizio 3

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

1. un metodo *verificaSomme* che riceve una matrice quadrata di double **m** e restituisce **true** se si ottiene lo stesso risultato sommando gli elementi della prima riga, sommando gli elementi sull'ultima colonna, e sommando quelli sulla diagonale principale, restituisce **false** altrimenti;
2. un metodo *costruisciVettore* che riceve una matrice quadrata di double **m** ed un double **k** e restituisce un vettore di double **v**. Gli elementi di **v** sono calcolati considerando solo le righe di **m** per cui la somma degli elementi è maggiore di **k**. In particolare, ogni elemento di **v** è pari alla somma degli elementi di un vettore riga che soddisfa la precedente condizione e gli elementi della diagonale secondaria di **m**. (si veda l'esempio);
3. un metodo *sottomatrice* che riceve una matrice quadrata di double **m** e restituisce una sottomatrice **sm** ottenuta a partire da **m** (ignorando l'ultima colonna se la dimensione è dispari). La sottomatrice **sm** deve contenere gli elementi di **m** i cui indici di riga e di colonna sono entrambi pari oppure sono entrambi dispari (si veda l'esempio);
4. un metodo *main* che legge una matrice quadrata di double **m** e invochi opportunamente i metodi descritti nei punti precedenti.

Esempio:

m =

7.1	4.0	2.3	1.1	1.2
1.2	0.0	1.1	5.1	2.0
7.2	0.3	3.1	4.2	5.2
1.3	2.3	1.3	0.0	1.8
6.3	2.1	3.2	4.1	5.5

1. *verificaSomme(m)* → *true*

2. *costruisciVettore(m, 15)* → [33.7 38.0 39.2]

(in particolare, la somma degli elementi della prima riga è maggiore di 15, e di conseguenza l'array risultante conterà in posizione 0 l'elemento 33.7 = 7.1+4.0+2.3+1.1+1.2+6.3+2.3+3.1+5.1+1.2)

3. *sottomatrice(m)* →

sm =

7.1	2.3
0.0	5.1
7.2	3.1
2.3	0.0
6.3	3.2