

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

Traccia G

Esercizio 1

Si consideri il seguente codice:

```
public static int metodoH (int m){
    int np = 0;
    for (int i = m; i > 0; i--){
        boolean t = true;
        for (int j=2; j<i; j++)
            if( i % j == 0)
                t = false;
        if(t)
            np += 1;
    }
    return np;
}
```

Si descriva sinteticamente la funzione svolta dal metodo e, in particolare, se ne mostri l'esecuzione nel caso in cui il valore del parametro sia **m=15**. Specificare quindi il contenuto dell'intero **np** restituito.

Esercizio 2

Scrivere un metodo *componivett* che riceve tre array **v1** e **v2** e **v3** di interi e ritorna un array **v4** della stessa lunghezza di **v3** in cui l'elemento di posizione **i** è calcolato come segue:

- se **v1[i]** è minore di **v2[i]**, **v4[i]** è uguale a **v3[i]**;
- altrimenti **v4[i]** è uguale alla somma degli ultimi **v2[i]** elementi di **v1**; cioè il valore contenuto in **v2[i]** indica quanti sono, partendo dall'ultimo, gli elementi del vettore **v1** da sommare; il risultato della somma va assegnato a **v4[i]**.

Ad esempio, se **v1** = [4, 3, 3, 9], **v2** = [2, 5, 1, 3] e **v3** = [3, -1, 0, -4], allora l'array **v4** restituito sarà [12, -1, 9, 15].

Esercizio 3

Si progetti una classe *GestioneMatrici* con i seguenti metodi:

- 1) un metodo *prodottoVicini* che, ricevuti in ingresso una matrice quadrata di interi **M** e una coppia di indici **i, j**, restituisce il prodotto degli elementi di **M** che si trovano intorno all'elemento in posizione **i, j**. Ad esempio, se il metodo viene invocato sulla matrice **M** sotto riportata con parametri **i=1, j=2**, esso restituisce la quantità 72, pari al prodotto dei valori colorati in grigio (2*(-1)*1*2*(-3)*1*2*3). Per gli elementi che sono sul bordo, i vicini sono ovviamente quelli possibili all'interno della matrice. Ad esempio se il metodo viene invocato con i parametri **i=4, j=2**, esso restituirà il valore -54.

		<i>j</i>					
		3	2	-1	1	4	⁰
	<i>i</i>	1	3	9	2	9	¹
		5	2	1	-3	1	²
		2	-3	3	1	0	³
		2	2	6	3	4	⁴
		0	1	2	3	4	

		<i>j</i>					
		3	2	-1	1	4	⁰
	<i>i</i>	1	3	9	2	9	¹
		5	2	1	-3	1	²
		2	-3	3	1	0	³
		2	2	6	3	4	⁴
		0	1	2	3	4	

- 2) un metodo *nuovaMatrice* che riceve in ingresso una matrice **M** di interi e restituisce una matrice **N** in cui gli elementi sono così ottenuti:
 - se il rapporto tra l'elemento **M[i][j]** ed il prodotto dei suoi vicini è maggiore o uguale ad 1, **N [i][j]** è pari a **M[i][j]**;
 - altrimenti, **N [i][j]** è pari al prodotto dei vicini di **M[i][j]**.
- 3) un metodo *main* in cui, una volta letta una matrice quadrata di interi **M**, e ottenuta tramite il metodo *nuovaMatrice* una nuova matrice **N**, viene ricavata e visualizzata la matrice **A** pari a **M + N**.