

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

**Traccia C**

**Esercizio 1**

Si consideri il seguente codice:

```
static int analizza( int vet[] ) {
    int j = 0, p = 0, l = vet.length;
    while( j < l / 2 ) {
        if ( vet[j] % vet[l-1-j] == 0 || vet[l-1-j] % vet[j] == 0 )
            p += vet[j];
        j++;
    }
    return p;
}
```

Si descriva sinteticamente il significato della funzione svolta dal metodo. Inoltre si mostri la traccia d'esecuzione ed il valore restituito dal metodo nel caso in cui il parametro sia **vet** = [10, 2, 4, 8, 3, 5].

**Esercizio 2**

Scrivere un metodo *CostruisciVettore* che riceve in ingresso un vettore *V* ed un intero *k*, e restituisce un vettore *VI* di dimensione *k* ottenuto inserendo ciclicamente in *VI* i quadrati degli elementi di *V*. Ad esempio, se *V*=[1,3,2] e *k*=7, il vettore restituito dal metodo è *VI*=[1<sup>2</sup>, 3<sup>2</sup>, 2<sup>2</sup>, 1<sup>2</sup>, 3<sup>2</sup>, 2<sup>2</sup>, 1<sup>2</sup>]=[1, 4, 9, 1, 9, 4, 1].

**Esercizio 3**

Si realizzi un programma Java per la gestione di matrici quadrate di numeri interi. Il programma deve in particolare prevedere dei metodi per svolgere i seguenti compiti:

1. Ricevere in ingresso una matrice *M* di interi, e restituire un vettore *R* di dimensione pari al numero di righe di *M*. *R* conterrà nella posizione *i* il valore medio dei valori non nulli contenuti nella riga *i* di *M*.
2. Ricevere in ingresso una matrice *M* di interi, e restituire un vettore *C* di dimensione pari al numero di colonne di *M*. *C* conterrà nella posizione *j* il valore massimo degli elementi non nulli contenuti nella colonna *j* di *M*.

Ad esempio, se *M* è la matrice mostrata in figura

$$M = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 6 & 0 & 1 & 7 \\ 0 & 1 & 3 & 0 \\ 1 & 2 & 9 & 0 \\ 4 & 6 & 3 & 3 \end{bmatrix} \end{matrix}$$

*R* sarà uguale a [4.66, 2, 4, 4] e *C* sarà uguale a [6, 6, 9, 7]

Realizzare infine un metodo *main* in cui venga letta da input una matrice di interi *M*, vengano invocati su *M* i metodi descritti ai punti 1 e 2 e venga verificato e segnalato in output se *C* è uguale ad *R*.