

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

Traccia A

Esercizio 1

Si consideri il seguente codice:

```
static boolean[] metodo(int[] x, int[] y)
{
    if( x.length != y.length ) return null;
    boolean[] n = new boolean[x.length];
    for(int i=0; i<n.length; i++)
        n[i] = false;
    int i=0;
    while( i < y.length )
    {
        if( (y[i]%x[x.length-1-i])== 0 )
            n[i]=true;
        i++;
    }
    return n;
}
```

Si descriva sinteticamente la funzione svolta dal metodo e, in particolare, se ne mostri l'esecuzione nel caso in cui il valore dei parametri sia $x=[3,2,5,4]$ e $y=[8,20,6,9]$. Specificare il contenuto dell'array **n** restituito.

Esercizio 2

Scrivere un metodo *public static void testprefisso(String s)* che, data una stringa *s*, stampi su output

- “prefisso corretto”, se la stringa inizia con una stringa *pref* e termina con questa stessa stringa *pref*, preceduta dal carattere '&';
- “prefisso scorretto”, se la precedente condizione non è soddisfatta;
- “prefisso non presente”, se il carattere '&' non è presente nella stringa *s*.

Ad esempio, se la stringa letta è “barbagianni&barba”, l'output sarà “prefisso corretto”.

Esercizio 3

Si deve realizzare un programma Java per la gestione di matrici quadrate di numeri reali. Il programma deve in particolare (a) estrarre da una matrice quadrata *A* di dimensione *N* (pari), quattro sottomatrici (*A11*, *A12*, *A21*, *A22*) ognuna di dimensione *N/2*:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

(b) calcolare la matrice $B=A_{11}*A_{22}-A_{12}*A_{21}$ e, infine, (c) verificare se la matrice *B* ottenuta ha gli elementi delle colonne di indice dispari tutti nulli. Strutturare il programma (almeno) nei seguenti metodi:

- *estrai* – che riceve una matrice quadrata *A* e due numeri interi *i* e *j* (che possono assumere i valori 1 e 2) e restituisce la sottomatrice *Aij*, cioè una delle sottomatrici *A11*, *A12*, *A21* e *A22* in cui può essere suddivisa la matrice *A*;
- *prodotto* – che riceve due matrici quadrate e restituisce il loro prodotto;
- *colonneDispariNulle* – che riceve una matrice e verifica (restituendo un booleano) se in essa sono tutti nulli gli elementi delle colonne di indice dispari;
- *main* - che legge una matrice *A* di numeri reali di dimensioni 12x12 e provvede, invocando opportunamente i metodi proposti, a realizzare i compiti di cui ai punti (a), (b) e (c), visualizzando su output la matrice *B* e l'esito della verifica (c).

Esercizio 4

Scrivere un metodo *verificasomma* che riceve due array *V1* e *V2* di *n* interi e ritorna *true* se la somma degli elementi di *V1* che si trovano nelle posizioni pari è uguale alla somma degli elementi di *V2* che si trovano nelle posizioni dispari, *false* altrimenti. La posizione 0 è da considerarsi pari. Ad esempio, se $V1 = [2,3,6,7,10]$ e $V2 = [10,11,2,7,6]$, allora il risultato sarà *true*.