

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

Traccia B

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 $x=[6, 3, 4, 7]$, $y=[14, 12, 6, 12]$. Specificare il contenuto dell'array n restituito.

Esercizio 2

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

- “suffisso corretto”, se la stringa termina con una stringa *suff* ed inizia con questa stessa stringa *suff*, terminata dal carattere '#’;
- “suffisso scorretto”, se la precedente condizione non è soddisfatta;
- “nessun suffisso”, se il carattere ‘#’ non è presente nella stringa s .

Ad esempio, se la stringa letta è “mente#superbamente”, l’output sarà “suffisso corretto”.

Esercizio 3

Si deve realizzare un programma Java per la gestione di matrici quadrate di numeri reali. Più in particolare, il programma deve (a) estrarre da una matrice quadrata A di dimensione N (pari), quattro sottomatrici (A_{11} , A_{12} , A_{21} , A_{22}) ognuna di dimensione $N/2$:

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

(b) calcolare la matrice $B=A_{11}*A_{21}-A_{12}*A_{22}$, (c) verificare se la matrice B ottenuta ha gli elementi delle righe di indice pari ordinati per valori crescenti. Si chiede di strutturare il programma (almeno) nei seguenti metodi:

Realizzare quindi (almeno) i seguenti metodi:

- *estrai* – che riceve una matrice quadrata A , due numeri interi r e c che individuano rispettivamente una riga e una colonna di A , un intero d che esprime una dimensione di sottomatrice, e restituisce la sottomatrice di A individuata da r , c e d . Ad esempio se $r=0$, $c=0$, $d=N/2$, allora la sottomatrice da estrarre è A_{11} , etc.;
- *prodotto* – che riceve due matrici quadrate e restituisce il loro prodotto;
- *righePariOrdinate* – che riceve una matrice e verifica (restituendo un booleano) se in essa gli elementi delle righe di indice pari sono disposti in ordine crescente;
- *main* - che legge una matrice A di numeri reali di dimensioni 12×12 e provvede, invocando opportunamente i metodi proposti, a realizzare i compiti di cui ai precedenti punti (a), (b) e (c) visualizzando, infine, il contenuto di B e un messaggio che indichi se B soddisfa la proprietà (c).

Esercizio 4

Scrivere un metodo *verificaprodotto* che riceve due array $V1$ e $V2$ di n interi e ritorna *true* se il prodotto degli elementi di $V1$ che si trovano nelle posizioni pari è uguale al prodotto degli elementi di $V2$ che si trovano nelle posizioni dispari, *false* altrimenti. La posizione 0 è da considerarsi pari. Ad esempio, se $V1 = [1,3,6,7,10]$ e $V2 = [12,20,2,3,17]$, allora il risultato sarà *true*.