

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

Traccia C

Esercizio 1

Si consideri il seguente codice:

```
public class Esercizio1{
    public static void main(String [] args)
    { int[] seq = {0,1,0,0,1,0};
      System.out.print(check(seq));
    }

    public static boolean check(int[] s)
    {
        int min;
        if (s.length <=1 || s[0] == 0)
            return false;
        else
            for (int i=1; i < s.length-1; i++)
                if (s[i]== s[i+1])
                    return true;
            return false;
        }
    }
}
```

Si descriva sinteticamente il metodo **check** e, in particolare, se ne mostri l'esecuzione nel caso specificato dal metodo **main**. Specificare cosa viene stampato.

Esercizio 2

Si scriva un metodo *gestionevett* che riceve un array **V1** di **n** interi e ritorna un vettore **V2** della stessa lunghezza in cui l'elemento in posizione **i**:

- è pari alla somma dell'elemento **V1[i]** e **V1 [i+1]** (se esiste, altrimenti si consideri **V1 [i+1] = 0**), se il valore **V1[i]** è positivo;
- è pari al valore assoluto del prodotto tra **V1[i]** e **V1 [i-1]** (se esiste, altrimenti si consideri **V1 [i-1] = 0**), se il valore **V1[i]** è negativo;

Ad esempio, se **V1** = [4, -3, 17, -10, -6], allora il risultato sarà: **V2** = [1, 12, 7, 170, 60]

Esercizio 3

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

1. un metodo *DiagonaleNonNulla* che restituisce *true* se almeno un elemento della diagonale principale ha un valore diverso da 0;
2. un metodo *Creavettore* che riceve una matrice quadrata di interi **A** e restituisce un vettore contenente tutti gli elementi presenti nelle due diagonali parallele alla diagonale principale nel triangolo al di sopra della diagonale principale. (Si proceda dall'alto verso il basso e da sinistra verso destra. Per maggiore chiarezza si veda l'esempio).
3. il metodo *main* che legge una matrice di interi **A** e provvede, invocando opportunamente i metodi proposti, a realizzare i compiti di cui ai punti (1) e (2).

Esempio:

A =	<table style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 0 10px;"><i>c</i>₀</td> <td style="padding: 0 10px;"><i>c</i>₁</td> <td style="padding: 0 10px;"><i>c</i>₂</td> <td style="padding: 0 10px;"><i>c</i>₃</td> <td style="padding: 0 10px;"><i>c</i>₄</td> </tr> <tr> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">0</td> <td style="border-bottom: 1px solid black; padding: 5px;">4</td> <td style="border-bottom: 1px solid black; padding: 5px;">3</td> <td style="border-bottom: 1px solid black; padding: 5px;">4</td> <td style="border-bottom: 1px solid black; padding: 5px;">0</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">5</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">5</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">10</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">10</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">5</td> <td style="padding: 5px;">4</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">4</td> <td style="padding: 5px;">5</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">0</td> <td style="padding: 5px;">7</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">7</td> <td style="padding: 5px;">0</td> </tr> </table>	<i>c</i> ₀	<i>c</i> ₁	<i>c</i> ₂	<i>c</i> ₃	<i>c</i> ₄	0	4	3	4	0	5	0	1	0	5	10	2	0	2	10	5	4	1	4	5	0	7	3	7	0
<i>c</i> ₀	<i>c</i> ₁	<i>c</i> ₂	<i>c</i> ₃	<i>c</i> ₄																											
0	4	3	4	0																											
5	0	1	0	5																											
10	2	0	2	10																											
5	4	1	4	5																											
0	7	3	7	0																											

- La funzione *DiagonaleNonNulla* invocata sulla matrice **A** restituisce *true*
- La funzione *Creavettore* invocata sulla matrice **A** restituisce *il seguente vettore*:
V=[4,1,2,5,3,0,10]