

Cognome e Nome

Matricola

Corso

 VECCHIO ORDINAMENTO
 NUOVO ORDINAMENTO

- NOTA BENE:**
1. **Non** è consentito l'uso di materiale didattico di qualsiasi natura.
 2. La presente traccia va **sempre** consegnata al termine della prova.

ESERCIZIO 1

Si descriva sinteticamente l'algoritmo implementato dal seguente metodo (e come esempio, si mostri un'esecuzione avente come input il vettore $v = [2,7,2,4,6,8]$ e l'intero $a = 10$).

```
public static int metodo1C (int v[], int a) {
    for (int k = 0; k < v.length-1; k++)
        if (v[k] + v[k+1] == a)
            return k;
    return -2;
}
```

ESERCIZIO 2

Si scriva un metodo **metodo2C** che riceve in input un vettore **V** e restituisce **true** se in **V** sono presenti due elementi **consecutivi** entrambi negativi o entrambi positivi.

Esempio: se $V = [2, -1, 8, 0, -3, 2]$ il metodo restituisce **false**; se $V = [1, -2, 3, 4, -1]$ il metodo restituisce **true**.

ESERCIZIO 3

Si realizzi una classe **Esercizio3C**, che contenga i seguenti metodi:

1. *confrontaRighe* che riceve una matrice **A** e restituisce **true** se la prima riga è uguale all'ultima.
2. *creaMatrice* che riceve in ingresso una matrice **A** (di dimensione $n*m$) e due interi $r < n$ e $c < m$ e restituisce la sottomatrice individuata dagli estremi $(0,0)$ ed (r,c) .
3. *controlla* che riceve in ingresso una matrice **A** ed un vettore **V** e restituisce **true** se ciascun elemento del vettore è maggiore di almeno un elemento nella matrice e **false** altrimenti.
4. *main* che invoca opportunamente i metodi ai punti 1,2 e 3.

Esempio:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 2 & 3 & 4 & 5 & 2 \\ 3 & 4 & 5 & 6 & 6 \\ 7 & 9 & 10 & 11 & 8 \\ 1 & 2 & 3 & 4 & 1 \end{bmatrix}$$

1. *confrontaRighe*(A) \rightarrow **true**

2. *creaMatrice*(A,1,2) \rightarrow

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

3. *controlla*(A,V) \rightarrow **true**

$$V = \begin{bmatrix} 3 & 2 & 10 & 11 \end{bmatrix}$$