

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

Traccia D

Esercizio 1

Si consideri il seguente codice:

```
public static int metodoD( int vet[] ) {
    if (vet.length == 0)
        return -1;
    else {
        int[] res = new int[vet.length];
        for (int i = 0; i < vet.length; i++){
            res[i] = 0;
            for (int j = 0; j <= i; j++)
                res[i] += vet[j];
        }

        int k = 0, z = 0;
        do {
            if (res[z] > 0)
                k++;
            z++;
        } while (z < res.length);
        return k;
    }
}
```

Si descriva sinteticamente la funzione svolta dal metodo e, in particolare, se ne mostri l'esecuzione nel caso in cui il valore del parametro sia **vet** = [2, 5, 2, -9, 3]. Specificare il valore restituito dal metodo.

Esercizio 2

Si progetti un metodo *comuni* che, ricevuti in ingresso due vettori **v1** e **v2** di interi, restituisce un reale **v** calcolato come rapporto tra gli elementi che appartengono a **v1** ma non a **v2**, ed il numero di elementi distinti che compaiono nei vettori. Ad esempio, se **v1** = [1 , 5 , 0, 4 , 13] e **v2** = [1 , 5 , 13 , 6], il valore restituito è 2/6: infatti gli elementi di **v1** che non appartengono a **v2** sono 0 e 4, mentre gli elementi distinti nei vettori sono 0, 1, 4, 5, 6, 13.

Esercizio 3

Si progetti una classe **GestioneMatrici** con i seguenti metodi:

- 1) un metodo *estraiDx* che riceve in ingresso una matrice **M** di interi e due interi *i* e *j* e restituisce la sottomatrice quadrata avente come vertice superiore sinistro l'elemento **M[i][j]** e comprendente tutti gli elementi che precedono (rispetto all'indice *j*) e seguono (rispetto all'indice *i*) l'elemento **M[i][j]**.
- 2) un metodo *estraiSx* che riceve in ingresso una matrice **M** di interi e due interi *i* e *j* e restituisce la sottomatrice quadrata avente come vertice inferiore destro l'elemento **M[i][j]** e comprendente tutti gli elementi che seguono (rispetto all'indice *j*) e precedono (rispetto all'indice *i*) l'elemento **M[i][j]**.

Ad esempio, se il metodo *estraiDx* è invocato con la matrice **A** e i valori *i* = 1 e *j* = 3, la sottomatrice sarà quella evidenziata in grigio in **A**. Se il metodo *estraiSx* è invocato con la matrice **B** e i valori *i* = 1 e *j* = 3, la sottomatrice sarà quella evidenziata in grigio in **B**.

		<i>j</i>					
		5	2	5	5	9	0
	<i>i</i>	1	3	5	2	7	1
A=		3	5	5	3	1	2
		2	7	5	1	0	3
		7	6	5	9	4	4
		0	1	2	3	4	

		<i>j</i>					
		5	2	5	5	9	0
	<i>i</i>	1	3	5	2	7	1
B=		3	5	5	3	1	2
		2	7	5	1	0	3
		7	6	5	9	4	4
		0	1	2	3	4	

- 3) Un metodo *main* in cui, una volta letta una matrice quadrata **M** di interi e un intero **k**, viene prima estratta la sottomatrice destra e poi la sottomatrice sinistra che parte dall'elemento **M[k][k]**, facendo uso dei metodi *estraiSx* ed *estraiDx*. Viene quindi stampato su output un messaggio che indica se la somma degli elementi della diagonale secondaria della prima sottomatrice è uguale alla somma degli elementi della diagonale secondaria della seconda sottomatrice.