

ESERCIZIO 1.

Scrivere un metodo iterativo **e1** con le seguenti caratteristiche:

- **e1** ha un parametro formale di tipo matrice bidimensionale di interi che puo' essere solo quadrata, o nulla.
- **e1** restituisce **true** quando:
 - la matrice non e' nulla e
 - la somma degli elementi di ciascuna riga coincide con la somma degli elementi della colonna corrispondente.

ESERCIZIO 2.

Scrivere un metodo ricorsivo dicotomico **e2** con le seguenti caratteristiche:

- **e2** ha un parametro formale di tipo matrice bidimensionale di interi che puo' essere solo quadrata, o nulla.
- **e2** restituisce **true** quando:
 - la matrice non e' nulla e
 - la somma degli elementi di ciascuna riga coincide con la somma degli elementi della colonna corrispondente.

Per il calcolo della somma degli elementi in una riga, definire un metodo ricorsivo **sommaR** co-variante. Per il calcolo della somma degli elementi in una colonna, definire un metodo ricorsivo **sommaC** contro-variante.

ESERCIZIO 3.

Siano dati:

- il metodo **parity**, qui sotto definito, da applicare esclusivamente ad un parametro attuale con almeno un elemento ($a.length \geq 1$)
- il predicato $P(i)$ seguente:

“Alla sua uscita, $parity(a,i)$ rende vero *per ogni* k . se $0 \leq k \leq i$, allora $a[k] = (k \% 2 == 0)$ ”.

1. Scrivere il predicato $P(0)$.

2. Scrivere il predicato $P(i-1) \implies P(i)$.
3. Dimostrare che $P(0)$ e' vero.
4. Dimostrare che $P(i-1) \implies P(i)$ e' vero, ragionando induttivamente.

```
static void parity(boolean[] a, int i) {  
    if (i < a.length) {  
        if (i == 0)  
            a[i] = true;  
        else {  
            parity(a, i - 1); //(A)  
            a[i] = !a[i - 1]; //(B)  
        }  
    }  
}
```

ESERCIZIO 4.

Disegnare lo stato della memoria immediatamente prima della disallocazione del record di attivazione del metodo stack, quando i ha valore 2.

```
static void stack(int[][] x, int i) {  
    if (i < x.length) {  
        int[] l = new int[x[i].length];  
        l[i] = x[i][i] + 1;  
        x[i] = l;  
        stack(x, i + 1); // (B)  
    }  
}  
  
public static void main(String[] args) {  
    int[][] y = {{0,0}, {0,0}};  
    stack(y, 0); // (A)  
}
```