

Simulazione.java

```
1 public class Simulazione {
2     /** ESERCIZIO 1.
3      * Scrivere un metodo iterativo e1 con le seguenti
4      * caratteristiche:
5      * -) e1 ha un parametro formale di tipo matrice
6      * bidimensionale di interi che puo' essere solo
7      * quadrata, o nulla.
8      * -) e1 restituisce true quando:
9      * a) la matrice non e' nulla e
10     * b) la somma degli elementi di ciascuna riga
11     * concide con la somma degli elementi della
12     * colonna corrispondente. */
13
14     /** ESERCIZIO 2.
15     * Scrivere un metodo ricorsivo dicotomico e2 con
16     * le seguenti caratteristiche:
17     * -) e2 ha un parametro formale di tipo matrice
18     * bidimensionale di interi che puo' essere solo
19     * quadrata, o nulla.
20     * -) e2 restituisce true quando:
21     * a) la matrice non e' nulla e
22     * b) la somma degli elementi di ciascuna riga
23     * concide con la somma degli elementi della
24     * colonna corrispondente.
25     * Per il calcolo della somma degli elementi in
26     * una riga, definire un metodo ricorsivo sommaR
27     * co-variante. Per il calcolo della somma degli
28     * elementi in una colonna, definire un metodo
29     * ricorsivo sommaC contro-variante. */
30
31
32
33
34
35
36
37
38
39
```

Simulazione.java

```
40  /** ESERCIZIO 3.
41  *  Siano dati:
42  *  -) il metodo parity, qui sotto definito,
43  *  da applicare esclusivamente ad un parametro
44  *  attuale con almeno un elemento (a.length>=1)
45  *  -) il predicato P(i) seguente:
46  *
47  *  "Alla sua uscita, parity(a,i) rende vero
48  *   'per ogni k.se 0<= k <= i, \
49  *   allora a[k]==(k%2==0)' ".
50  *
51  *  1) Scrivere il predicato P(0).
52  *  2) Scrivere il predicato P(i-1) ==> P(i).
53  *  3) Dimostrare che P(0) e' vero.
54  *  4) Dimostrare che P(i-1) ==> P(i) e' vero,
55  *  ragionando induttivamente. */
56  static void parity(boolean[] a, int i) {
57      if (i < a.length) {
58          if (i == 0)
59              a[i] = true;
60          else {
61              parity(a, i - 1); //(A)
62              a[i] = !a[i - 1]; //(B)
63          }
64      }
65  }
66
67
68
69
70
71
72
73
74
75
76
77
78
```

Simulazione.java

```
79  /** ESERCIZIO 4. Disegnare lo stato della  
80  * memoria immediatamente prima della  
81  * deallocazione del record di attivazione  
82  * del metodo stack, quando i ha valore 2. */  
83  static void stack(int[][] x, int i) {  
84      if (i < x.length) {  
85          int[] l = new int[x[i].length];  
86          l[i] = x[i][i] + 1;  
87          x[i] = l;  
88          stack(x, i + 1); // (B)  
89      }  
90  }  
91  
92  public static void main(String[] args) {  
93      int[][] y = {{0,0}, {0,0}};  
94      stack(y, 0); // (A)  
95  }  
96 }
```