

Risolvi i seguenti esercizi giustificando sinteticamente le risposte.

1. Scheme e Java

Traduci la procedura Scheme definita a lato in un corrispondente metodo in Java, basato sulla stessa struttura ricorsiva.

```
(define tiles
  (lambda (d k) ; 0 <= d < 4, k >= 0
    (if (= k 0)
        0
        (+ (* 2 (tiles d (- k 1)))
           (tiles (remainder (+ d 3) 4) (- k 1))
           (tiles (remainder (+ d 1) 4) (- k 1))
           (if (> d 0) 0 1) )
        )))
```

2. Memoization

Trasforma la soluzione dell'esercizio 1 in un programma corrispondente, formalizzato nel linguaggio *Java*, che applichi la tecnica di memoization.

3. Astrazione procedurale

La procedura con valori procedurali *ht* è definita in modo che valutando le espressioni `(let ((f (ht x))) (f y))` e `(tiles x y)` il risultato sia lo stesso. Come indicato nell'esercizio 1, si assume che *x* sia un intero compreso nell'intervallo $[0, 3]$ e che *y* sia un numero naturale. Completa il codice Scheme della procedura *ht* riportato qui sotto. Il corpo della procedura può fare riferimento ricorsivo ad *ht*, ma non può chiamare la procedura *tiles* dell'esercizio 1.

```
(define ht
  .....
  .....
  (if (= k 0)
      0
      (+ (* 2 ..... )
         .....
         .....
         (if (> d 0) 0 1) )
      ) ... ) ; chiusura di tutte le parentesi
```

4. Dimostrazioni per induzione

Con riferimento alla procedura *tiles* dell'esercizio 1, assumi che per tutti i valori di *n* nell'insieme dei naturali siano già state dimostrate le seguenti proprietà:

$$(tiles\ 1\ n+1) \rightarrow (4^n - 1)/3, \quad (tiles\ 2\ n+1) \rightarrow (4^n - 1)/3 - 2^n + 1 \quad \text{e} \quad (tiles\ 3\ n+1) \rightarrow (4^n - 1)/3$$

Dimostra quindi per induzione che anche la seguente proprietà vale per ogni *n* naturale:

$$(tiles\ 0\ n+1) \rightarrow (4^n - 1)/3 + 2^n$$

In particolare:

- Scrivi formalmente la proprietà che intendi dimostrare per induzione.
- Scrivi formalmente la proprietà che esprime il caso base.
- Scrivi formalmente l'ipotesi induttiva.
- Scrivi formalmente la proprietà che occorre dimostrare come passo induttivo.
- Dimostra formalmente il caso base.
- Dimostra formalmente il passo induttivo.

5. Classi in Java

Con riferimento all'esercizio 1, il seguente metodo statico in Java è definito in modo tale che `tiles4(y).comp(x)` assuma lo stesso valore di `(tiles x y)` per ogni *x* compreso nell'intervallo $[0, 3]$ e per ogni *y* naturale.

```
public static TilesVector tiles4( int k ) { // k >= 0
  if ( k == 0 ) {
    return new TilesVector(0,0,0,0);
  } else {
    TilesVector t = tiles4(k-1);
    return new TilesVector( 2*t.comp(0)+t.comp(1)+t.comp(3)+1,
                           2*t.comp(1)+t.comp(2)+t.comp(0),
                           2*t.comp(2)+t.comp(3)+t.comp(1),
                           2*t.comp(3)+t.comp(0)+t.comp(2) );
  }
}
```

Formalizza in Java una classe *TilesVector* per rappresentare quaterne, compatibilmente con l'uso che se ne fa nel codice riportato sopra.