

Risolvi i seguenti esercizi giustificando sinteticamente le risposte.

1. Scheme e Java

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

```
(define gf
  (lambda (n k)
    ; 1 <= k <= n
    (if (even? n) ; n pari
        (- (* 2 (gf (quotient n 2) (quotient (+ k 1) 2)))
           (remainder k 2))
        (if (= n 1)
            1
            (let ((x (gf (- n 1) (+ (remainder k (- n 1)) 1))))
              (if (> x k) (+ x 1) x)))))))
  )))
```

2. Programmazione dinamica

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

3. Ricorsione di coda

Trasforma la procedura *mirror* definita qui sotto in un programma che applica la ricorsione di coda al posto della ricorsione generale, mantenendo la stessa logica risolutiva.

```
(define mirror
  (lambda (x) ; x stringa
    (let ((n (string-length x)))
      (cond ((= n 0) null)
            ((= n 1) (list (string-ref x 0)))
            ((char=? (string-ref x 0) (string-ref x (- n 1)))
             (append (mirror (substring x 1 (- n 1))) (list (string-ref x 0))))
            (else (mirror (substring x 1 (- n 1)))))))))
```

4. Dimostrazioni per induzione

Con riferimento alla procedura *gf* dell'esercizio 1, dimostra per induzione che, data qualunque coppia di numeri naturali $i \geq 0$, $v \in [1, 2^i]$, il risultato della valutazione dell'espressione $(gf\ 2^i\ v)$ è v . 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

Formalizza in Java una classe *PlaneVect* per rappresentare vettori nel piano. Per ogni oggetto u della classe *PlaneVect* deve essere definito il seguente protocollo: un costruttore; il metodo $u.length()$ che consente di conoscere la lunghezza del vettore; il metodo $u.orientation()$ per determinare l'angolo fra la direzione del vettore e quella di un asse orizzontale di riferimento; il metodo $u.add(v)$ che restituisce la somma vettoriale di u e v ; il metodo $u.rescale(c)$ che restituisce un vettore con la stessa direzione di u , ma di lunghezza riscalata di un fattore c ; il metodo $u.rotate(a)$ che restituisce un vettore con la stessa lunghezza di u , ma ruotato di a radianti in verso antiorario.