

Corso di Programmazione

Esame del 3 Luglio 2001

cognome e nome

Risolvi i seguenti esercizi e riporta le soluzioni in modo chiaro nei fogli protocollo, giustificando sinteticamente le soluzioni proposte.

1. Definizione di procedure in Scheme

Scrivi una procedura *i-list* in Scheme che, data una funzione f definita sull'insieme dei naturali e a valori naturali, e dati due numeri naturali x e n , assume come valore la lista delle prime n applicazioni iterate di f ad x , cioè la lista dei valori:

$$(f(x), f^{(2)}(x), f^{(3)}(x), \dots, f^{(n)}(x)) = (f(x), f(f(x)), f(f(f(x))), \dots, f(f(\dots f(x)\dots)))$$

Quindi, scrivi un'espressione (*i-list* ...) che, applicando la procedura *i-list* ad opportuni argomenti, consenta di generare la lista dei primi 100 numeri naturali positivi.

2. Dimostrazioni per induzione

Con riferimento all'esercizio precedente, dimostra per induzione che l'ultimo elemento della lista che si ottiene valutando (*i-list* f x n) per $n > 0$ è $f^{(n)}(x)$. 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 si deve dimostrare come passo induttivo.
- Dimostra formalmente il caso base.
- Dimostra formalmente il passo induttivo.

3. Classi in Java

Definisci una classe in Java che permetta di istanziare liste di interi. Tale classe deve prevedere un costruttore per rappresentare la lista vuota e i seguenti metodi:

- *cons(x)* che aggiunge l'intero x all'inizio di una lista;
- *car()* che restituisce il primo elemento di una lista non vuota;
- *cdr()* che elimina il primo elemento da una lista non vuota;
- *null_list()* che verifica se una lista è vuota.

4. Programmi iterativi in Java

Utilizzando istanze della classe definita nell'esercizio precedente, scrivi un frammento di programma iterativo in Java che rovesci una lista data (cioè che costruisca una lista con gli stessi elementi, ma in ordine inverso).

5. Invarianti

Completa il seguente frammento di programma in Java per calcolare la fattorizzazione di un numero naturale $n > 2$: al termine dell'esecuzione i fattori primi e i corrispondenti esponenti sono rappresentati dalle componenti $p[i]$ e, rispettivamente, $e[i]$ dei vettori p ed e , per $i \in [0, k]$. Inoltre, descrivi sinteticamente (in modo informale) quali potrebbero essere le asserzioni invarianti del comando *while* più esterno che ne dimostrano la correttezza.

```
x = n;  q = _____ ;  j = 0;
while ( x > 1 ) {
    k = j;  e[k] = 0;  p[k] = q;
    while ( _____ ) {
        e[k] = e[k]+1;  x = _____ ;  j = k+1;
    }
    q = q+1;
}
```