

# Corso di Programmazione

Esame del 24 Luglio 2002

cognome e nome

Risolvi i seguenti esercizi e riporta le soluzioni in modo chiaro su questo foglio, giustificandole sinteticamente.

## 1. Definizione di procedure in Scheme

Scrivi una procedura in Scheme che, date due liste di interi  $L1$  e  $L2$ , assume come valore la lunghezza della più lunga parte iniziale di  $L1$  che è anche una parte della lista  $L2$ , composta da elementi consecutivi e disposti nello stesso ordine. In altri termini, interessa la lunghezza del più lungo prefisso di  $L1$  che è una sottosequenza di  $L2$ .

Per esempio, se  $L1 = (1\ 3\ 8\ 5\ 6\ 7)$  e  $L2 = (1\ 3\ 4\ 8\ 1\ 3\ 8\ 5\ 3\ 8\ 5\ 6\ 7)$ , il valore che deve essere calcolato è 4, cioè la lunghezza della sottosequenza comune evidenziata in grassetto.

## 2. Dimostrazioni per induzione

Formalizza in Scheme una procedura che, dato un numero naturale positivo  $n$ , assume come valore l'esponente  $e$  di 2 nella fattorizzazione di  $n$  in fattori primi.

Quindi, dimostra per induzione che il programma è corretto, cioè che  $n$  è divisibile per  $2^e$  ma non per  $2^{e+1}$ ; 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 bidirezionali* di interi, in cui risulta individuato un particolare elemento di riferimento. Tale classe deve prevedere un costruttore per rappresentare la lista bidirezionale di un solo elemento e i seguenti metodi:

- `ref()` che restituisce il valore dell'elemento di riferimento;
- `leftmost()` che verifica se il riferimento è l'elemento più a sinistra;
- `rightmost()` che verifica se il riferimento è l'elemento più a destra;
- `left()` che sposta il riferimento a sinistra di un elemento;
- `right()` che sposta il riferimento a destra di un elemento;
- `addLeft(x)` che aggiunge l'intero  $x$  a sinistra del riferimento;
- `addRight(x)` che aggiunge l'intero  $x$  a destra del riferimento.

## 4. Programmi iterativi in Java

Utilizzando istanze della classe definita nell'esercizio precedente, scrivi un frammento di programma iterativo in Java, esterno alla classe, per inserire al posto giusto, cioè in modo da preservare l'ordinamento degli elementi, un nuovo intero  $n$  in una lista bidirezionale che si suppone già ordinata in ordine crescente da sinistra a destra. Il riferimento può essere ovunque al momento in cui si esegue l'operazione.

## 5. Invarianti

Definisci un metodo statico in Java che, dato un numero naturale positivo  $n$  e dato un numero primo  $p$ , assume come valore l'esponente di  $p$  nella fattorizzazione di  $n$  in fattori primi. Inoltre, descrivi sinteticamente quali potrebbero essere le asserzioni invarianti dei comandi iterativi utilizzati che ne dimostrano la correttezza. Usa la seguente intestazione:

```
public static int esponenteFattorizzazione( int n, int p ) {  
    . . .  
    . . .  
}
```