

# Corso di Programmazione

Esame del 5 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 *comb-list* in Scheme che, data una funzione  $f: \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$ , definita per tutte le coppie di elementi del dominio  $\mathbf{D}$  e a valori in  $\mathbf{D}$ , e dati un elemento  $x \in \mathbf{D}$  e un numero naturale positivo  $n$ , assuma come valore la lista:

$$(x, f(x,x), f(f(x,x),x), f(f(f(x,x),x),x), \dots, f(f(f\dots f(x,x)\dots,x),x),x))$$

dove nell'ultimo elemento  $f$  è applicata  $n-1$  volte. Quindi, scrivi un'espressione (*comb-list* ...) che, applicando la procedura *comb-list* ad opportuni argomenti, consenta di generare la lista dei valori delle prime 20 potenze di due.

## 2. Dimostrazioni per induzione

Con riferimento all'esercizio precedente, dimostra per induzione che l'ultimo elemento della lista che si ottiene valutando (*comb-list string-append "a" n*) per  $n > 0$  è la stringa "aa ... a", composta da  $n$  lettere  $a$ , che può essere denotata sinteticamente da  $a^n$ . 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 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 circolari* (ad anello) di interi, in cui risulta individuato un particolare elemento di riferimento. Tale classe deve prevedere un costruttore per rappresentare la lista circolare di un solo elemento (che quindi è il successore di se stesso) e i seguenti metodi:

- *ref()* che restituisce il valore dell'elemento di riferimento;
- *length()* che restituisce il numero di elementi della lista ciclica.
- *next()* che sposta il riferimento al successivo elemento;
- *add(x)* che aggiunge l'intero  $x$  fra l'elemento di riferimento e quello successivo.

## 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 duplicare una lista ciclica (cioè per creare una copia con gli stessi elementi e con lo stesso ordine ciclico; non è rilevante, invece, quale sia l'elemento di riferimento).

## 5. Invarianti

Definisci un metodo statico in Java per verificare se due numeri naturali positivi sono primi fra loro, cioè non hanno divisori comuni maggiori di uno. Inoltre, descrivi sinteticamente quali potrebbero essere le asserzioni invarianti dei comandi iterativi utilizzati che ne dimostrano la correttezza. Usa la seguente intestazione:

```
public static boolean primiFraLoro( int x, int y ) {  
    . . .  
    . . .  
}
```