

Corso di Linguaggi di Programmazione

QUINTO APPELLO A.A. 2006/2007

C. Braghin e A. Ceselli

15 Gennaio 2008

Cognome e Nome:

Numero matricola:

Domanda 1. Si discuta brevemente il concetto di *tipo* nei linguaggi di programmazione. In particolare, si discutano brevemente almeno i seguenti punti (con esempi su di uno o più linguaggi a piacere): (i) definizione di tipo; (ii) tipo statico vs tipo dinamico; (iii) tipizzazione esplicita vs implicita; (iv) tipizzazione forte vs debole; (v) linguaggio type safe vs linguaggio type unsafe.

Domanda 2. Discutere il concetto di Abstract Data Type (ADT). Indicare ed analizzare le finalità, le politiche di sviluppo e le caratteristiche implementative di un ADT. Riportare un esempio di ADT, discutendone e confrontandone l'implementazione in almeno due linguaggi di programmazione.

Domanda 3. Scrivere in Scheme una funzione `sottolista?` che prende come argomento due liste `sub-list` e `list` di interi (opzionale: controllare che le liste siano effettivamente formate solo da interi) e ritorna `#t` se `sub-list` è sottolista di `list`, altrimenti ritorna `#f`. Per definizione la lista vuota è sottolista di una qualsiasi lista.

La funzione `sottolista?` si dovrebbe comportare nel modo seguente:

```
> (sottolista? '(2) '(1 2 3))
#t
> (sottolista? '(2 3) '(1 2 3))
#t
> (sottolista? '(2 3 4) '(1 2 3))
#f
> (sottolista? '(3) '(1 2 3))
#t
> (sottolista? '() '(1 2 3))
#t
> (sottolista? '() '())
#t
```

Domanda 4. Realizzare un programma Prolog *interleave(X,Y,R)*, che verifichi che la lista R possa essere ottenuta partendo dalle due liste X e Y, considerando in ordine il primo elemento di X, il primo elemento di Y, il secondo elemento di X, il secondo elemento di Y e così via. Esempio:

```
?- interleave([a,b,c],[1,2,3],[a,1,b,2,c,3]).
```

Yes

```
?- interleave([a,b,c],[1,2,3],[a,1,2,b,c,3]).
```

No

Se una delle liste X o Y è più lunga dell'altra, la porzione in eccedenza dovrà comparire in coda ad R. Esempio:

```
?- interleave([a,b,c],[1],[a,1,b,c]).
```

Yes

```
?- interleave([a,b,c],[1],[a,1]).
```

No

Si considerino anche i seguenti esempi:

```
?- interleave([a,b,c],[1,2,3],R).
```

R = [a, 1, b, 2, c, 3]

Yes

```
?- interleave([a,b,c],Y,[a,1,b,2,c,3]).
```

Y = [1, 2, 3]

Yes

```
?- interleave(X,[1,2,3],[a,1,b,2,c,3]).
```

X = [a, b, c]

Yes

e si discuta l'invertibilità del predicato.