

Corso di Linguaggi di Programmazione

SECONDO APPELLO A.A. 2006/2007

C. Braghin e A. Ceselli

11 Luglio 2007

Cognome e Nome:

Numero matricola:

Domanda 1. Si discuta della forma BNF come metalinguaggio per descrivere formalmente la sintassi di un linguaggio di programmazione. In particolare, si considerino brevemente almeno i seguenti punti: (i) differenza tra simboli terminali e non terminali, (ii) esempio di definizione in BNF di un tipo di istruzione di uno dei linguaggi visti a lezione (mostrando la produzione di una specifica istruzione), (iii) problema dell'ambiguità di una grammatica.

Domanda 2. Si consideri il problema della sincronizzazione degli accessi ad un buffer condiviso. Quali tipi di sincronizzazione devono essere garantiti? Si descrivano le caratteristiche di due opportuni meccanismi di sincronizzazione, indicando il livello di granularità del codice a cui operano ed evidenziandone reciproci vantaggi e svantaggi. Supportare l'argomentazione anche attraverso esempi in diversi linguaggi di programmazione.

Domanda 3. Scrivere in Scheme una funzione `ordine_crescente?` che prende come argomento una lista di interi (opzionale: controllare che la lista sia effettivamente formata solo da interi) e ritorna `#t` se ciascun numero della lista è maggiore o uguale del numero che lo precede, altrimenti ritorna `#f`. Ad esempio, la lista (1 2 3 5 7 9 9 11 11) è in ordine crescente, mentre (1 2 3 5 7 9 9 11 1) non lo è. Per definizione la lista vuota è in ordine crescente.

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

```
> (ordine_crescente? '(2 3 3 5 7 9 9 10) )  
#t
```

```
> (ordine_crescente? '(1 2 3 5 7 9 9 11 1) )  
#f
```

```
> (ordine_crescente? '(17) )  
#t
```

```
> (ordine_crescente? '() )  
#t
```

Domanda 4. Realizzare un programma Prolog che conti il numero di occorrenze di una costante in una lista: `count(El,Lista,R)` :- *R* è il numero di occorrenze di *El* in *Lista*. Il predicato è invertibile?