

# Corso di Linguaggi di Programmazione

## TERZO APPELLO A.A. 2007/2008

C. Braghin e A. Ceselli

19 Settembre 2008

Cognome e Nome: .....

Numero matricola: .....

**Domanda 1.** Si discuta l'utilità di un formalismo come la BNF per la descrizione di linguaggi di programmazione, specificando la differenza tra la sintassi e la semantica di un linguaggio. Giustificare la risposta. Si chiarisca inoltre perché la forma BNF non possa venire utilizzata per descrivere anche la semantica dei linguaggi, specificando e descrivendo anche mediante esempi l'alternativa comunemente usata.

**Domanda 2.** Elencare e spiegare brevemente quali sono le features richieste affinché un linguaggio di programmazione supporti la programmazione orientata agli oggetti. Discutere il concetto di ereditarietà e le sue possibili implementazioni nei linguaggi di programmazione. Analizzare le relazioni tra ereditarietà e scoping nella programmazione orientata agli oggetti.

**Domanda 3.** Scrivere in Scheme una funzione `split` che prende come argomento una lista di interi e ritorna come risultato una lista che contiene tre liste: la prima contenente tutti i numeri negativi della lista in input, la seconda tutti gli zeri e la terza tutti i numeri positivi (mantenendo le eventuali occorrenze multiple di un numero). Ad esempio:

```
> (split '(-1 2 3 4 0) )
((-1) (0) (2 3 4))
> (split '(2 4 6) )
(()) (2 4 6)
> (split '(-2 -4 -6) )
((-2 -4 -6) () ())
> (split '() )
(()) () ()
```

Opzionale: verificare che la lista in input sia effettivamente costituita solo da interi.

**Domanda 4.** Realizzare un programma Prolog *permutation(X,Y)* che data in input una lista X, ad esempio contenente numeri interi, restituisca una lista Y ottenuta permutando gli elementi di X:

Esempio:

```
?- permutation([1,2,3],Y) .
```

```
Y = [2, 1, 3]
```

Yes

```
?- permutation([a,b,a],Y) .
```

```
Y = [a, a, b]
```

Yes

E' possibile utilizzare il programma per generare *tutte* le possibili permutazioni? Come?

Si consideri infine la seguente interrogazione

```
?- permutation([1,2,3],[2,3,1]) .
```

indicando, motivando e discutendo l'esito atteso da questa query.

**Suggerimento:** partire dall'implementazione del predicato *select(X,Y,Z)* che, dato in input un elemento X ed una lista Y, restituisca una lista Z identica alla lista Y, ma in cui un'occorrenza dell'elemento X è stata cancellata.