

Linguaggi di Programmazione

<i>Nome e Cognome</i>	
<i>Matricola</i>	
<i>Anno di corso</i>	
<i>Telefono</i>	

1. Specificare la grammatica BNF di un linguaggio in cui ogni frase corrisponde ad una o più dichiarazioni di variabili. Ogni dichiarazione (che, ad eccezione dell'ultima, è separata dalla successiva mediante una virgola) è definita dal nome della variabile e dal suo tipo. Possibili tipi sono **int**, **real**, **string**, **record** e **function**. I primi tre tipi sono semplici. Il tipo **record** (terminato dalla keyword **end**) definisce una struttura i cui campi sono a loro volta caratterizzati da identificatori e relativi tipi. Il tipo **function** esprime il prototipo di una funzione avente uno o più parametri in ingresso ed un tipo in uscita preceduto dalla keyword **return**. Il linguaggio è perfettamente ortogonale, nel senso che le dichiarazioni sia di record che di funzioni possono mutuamente innestarsi senza limiti di profondità (ad esempio, il campo di un record può essere una funzione). Ecco un esempio di frase contenente sette dichiarazioni:

```

alfa: int,
beta: real,
gamma: string,
R: record
    A: string,
    B: record C: real, D: int end,
    E: string
end,
F: function (X: int, Y: string) return string,
G: function (A: real, B: function (C: string) return int) return record A: int, B: real end,
H: record
    N: string,
    M: function (C: int) return function (D: string) return int,
    L: real
end

```

NB: Non è richiesta la specifica degli identificatori (considerati terminali).

2. Data la seguente BNF relativa alla descrizione di numeri in base cinque:

```

pentanum → pentanum 0 | penanum 1 | pentanum 2 | pentanum 3 | pentanum 4
          0 | 1 | 2 | 3 | 4 |

```

definire la semantica denotazionale che esprime il valore di una frase (sequenza di cifre).

3. Dopo aver specificato la funzione `member` in *Haskell* (appartenenza di un elemento ad una lista), definire la funzione booleana `is_square`, che stabilisce se un numero naturale in ingresso è il quadrato di un altro numero naturale.

4. Supponendo di avere una base di fatti *Prolog* relativa alla specifica di un relazione genitore-figlio, come nel seguente esempio,

```
genitore(guido, elena).  
genitore(guido, luisa).  
genitore(elena, giovanni).  
genitore(elena, paola).  
genitore(luisa, andrea).  
genitore(luisa, dario).  
  
...
```

specificare in *Prolog* i seguenti predicati:

- a. fratello(X, Y): X è fratello di Y;
- b. cugino(X, Y): X è cugino di Y;

5. Illustrare come si risolve il problema della valutazione delle espressioni aritmetiche partendo dalla tabella delle precedenze/associatività.
6. Discutere la differenza tra tipi di dati astratti parametrizzati in Ada e C++.