Corso di Paradigmi di Programmazione
Prova scritta del 5 luglio 2005.

Tempo a disposizione: ore 2.

1. Si consideri il seguente frammento di codice (notazione: `^elemento` è il tipo dei puntatori a `elemento`; `p^` è la dereferenziazione del puntatore `p`):

```
type lista = ^elemento;
type elemento = record
                valore: intero;
                prossimo: lista
                end
....
p := lista;
while (p <> nil ) and (p^.valore <> 3) do
        p := p^.prossimo;
```

   Sotto quali ipotesi tale frammento è corretto ? Motivare la risposta.

2. Si dica, motivando la risposta, quali delle seguenti regole (o produzioni) possono comparire in una grammatica libera da contesto (i simboli maiuscoli sono non-terminali, quelli minuscoli sono terminali):

$$A \to B, \quad a \to A, \quad AB \to aA, \quad A \to AaA, \quad aB \to aA$$

3. Si dica cosa viene stampato dal seguente frammento di codice in un linguaggio con scoping statico e passaggio dei parametri per valore (con `void f(value int A[10])` si intende la dichiarazione di una funzione con parametro formale per valore di tipo array di 10 interi):

```
{int z = 0;
 int j;
 int V[10];
 void f(value int A[10]){
    int i;
    for (i=0; i<10; i=i+1){
       A[i] = z;
       z = z + A[i];
       }
    }
 for (j=0; j<10; j=j+1)
    V[j] = 1;
 f(V);
 write(z, V[9]);
}
```

4. Si consideri il seguente frammento in uno pseudolinguaggio con modello delle variabili a riferimento e garbage collector mediante contatori dei riferimenti; se `OGG` è un generico oggetto nello heap, indichiamo con `OGG.cont` il suo contatore (nascosto).

```
class C { int n; C next;}
C foo(){                                                    2
   C p = new C();      // oggetto OGG1
   p.next = new C();   // oggetto OGG2                     4
   C q = new C();      // oggetto OGG3
   q.next = p.next;                                         6
   return p.next;
   }                                                        8
C r = foo();
```

Si dica quali sono i valori dei contatori dei riferimenti dei tre oggetti dopo l'esecuzione della linea 6 e della linea 9.

5. Si consideri il comando case

```
case Exp  of
        1:      C1;
        2,3:    C2;
        4..6:   C3;
        7:      C4;
        else:   C5
```

e si fornisca uno pseudo-codice assembly, efficiente, corrispondente alla traduzione di un tale comando.

6. Si consideri la dichiarazione di array multidimensionale `int A[10][10][10]`. Sappiamo che: un intero è memorizzato su 4 byte; l'array è memorizzato in ordine di riga, con indirizzi di memoria crescenti (cioè se un elemento è all'indirizzo i, il successivo è a $i+4$ ecc.) Qual è l'offset dell'elemento `A[2][2][5]` rispetto all'elemento `A[0][0][0]`? (Si risponda in notazione decimale).

7. In uno pseudolinguaggio con eccezioni viene data la seguente definizione di funzione:

```
int f(int x){
   if (x==0) return 1;
   else if (x==1) throw X;
        else if (x==2) return f(1);
             else try {return f(x-1);} catch X {return x+1;}
   }
```

Quale valore viene ritornato da `f(4)`?

8. **Solo per il corso AL** Si consideri il seguente programma logico

```
member (X, [X| Xs]).
member(X,  [Y|Xs]):- member(X, Xs).
```

dove il costruttore `[X | Xs]` indica la lista con testa X e coda Xs. Si dica quale è il risultato della valutazione del goal `member(f(X), [ 1, f(2), 3] )` (dove `[ 1, f(2), 3]` indica la lista che contiene i tre elementi `1, f(2)` e `3`.

9. **Solo per il corso MZ** È noto che Scheme adotta una strategia di valutazione call-by-value. Date le seguenti definizioni:

```
K = (lambda (x) (lambda (y) (x)))
I = (lambda (x) (x))
Delta = (lambda (x) (x x))
```

Qual è il risultato della valutazione dell'espressione `((K (I 3)) (Delta Delta))` ?