Corso di Paradigmi di Programmazione
Prova scritta del 29 maggio 2012.

Tempo a disposizione: ore 2.
Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Con la notazione $\mathcal{C}^{L}_{L_1,L_2}$ indichiamo un compilatore da L_1 a L_2 scritto in L. Con $\mathcal{I}^{L}_{L_1}$ indichiamo un interprete scritto in L per il linguaggio L_1. Si dica se la seguente espressione ha senso

$$\mathcal{I}^{L}_{L_1}(\mathcal{C}^{L_1}_{L_1,L_2},\mathcal{C}^{L_1}_{L_1,L_2})$$

   Se la risposta è "no" si motivi tale fatto; altrimenti si dica qual è il risultato ottenuto.

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. Un certo linguaggio è basato su questi tre principi: (i) *scoping statico*; (ii) *dichiarazioni implicite*: la prima volta in un blocco che un nome compare a sinistra dell'operatore di assegnamento (`=`) viene creata una nuova associazione per quel nome (legato al valore dell'espressione a destra dell'assegnamento); (iii) tutte le associazioni create in una funzione sono *locali* a quella funzione. Supponendo di adottare questi tre principi, si dica cosa stampa il seguente frammento (espresso nel solito pseudolinguaggio):

```
A = 10;
B = 1;
void foo(){
    B = A+1;
    write(B);
}
void fie(){
    A = 5;
    foo();
}
fie();
write(A,B);
```

4. Si considerino le seguenti dichiarazioni in Java:

```
class A{
   int x = 5;
   int f(int n){return n+1;}
}
class B extending A{
   int x = 2;
   int y = 2;
   int f(int n){return x+1;}
   }
A a = new B();
```

   Nello scope di tali dichiarazioni, qual è il valore dell'espressione `a.f(0)` ?

5. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome e scope dinamico

```
int x = 10;
void foo(name int y){
    x = x++ + 1;
    y = y++ + 10;
    x = x+y;
    write(x);
}
{int x = 50;
 foo(x);
 write(x);
}
```

6. Si dica, motivando la risposta, se un linguaggio con allocazione statica della memoria può contenere un comando di iterazione indeterminata.

7. Si consideri il seguente frammento in uno pseudolinguaggio con parametri di ordine superiore:

```
{void foo (int f(), int n){
    int m = 10;
    int fie(){
      write(n,m);
    }
    if (n==0) {int n = 20;
                    f();
                    }
    else {m = 30;
              foo(fie,0);
              }
 }
 int g(){
    write(10);
 }
 int m = 100;
 foo(g,1);
}
```

   Si dica cosa stampa il frammento con con scope dinamico e shallow binding.

8. L'esecuzione del seguente frammento di codice su una certa implementazione risulta nella stampa dei valori 4 e 1.

```
int W[10];
int x = 4;
for (int i=0, i<10, i++) W[i]=i;
void foo(int x;  int y){
x = x+1;
y=1;
}
foo (x, W[x])
write (W[4])
write  (W[5])
```

   Si fornisca una possibile spiegazione.