Corso di Paradigmi di Programmazione
Prova scritta del 9 Giugno 2004.

Tempo a disposizione: ore 2.

1. Si spieghi sinteticamente cosa è la pila di sistema e a che cosa serve.

2. Si definisca un grammatica che generi tutte le possibili sequenze di parentesi graffe bilanciate.

3. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping statico e passaggio di parametri per riferimento. La primitiva `write(x)` permette di stampare un valore intero.

```
{int x = 2;

void pippo(reference int y){
     x = x + y;
     y = y + 1}

   { int x = 5;
     int y = 5;
     pippo(x);
     write(x);
   }

write(x); }
```

4. QEUSTO E' DA CAMBIARE

   Si consideri il seguente frammento di codice in uno pseudo-linguaggio che ammetta passaggio dei parametri per riferimento e per nome.

```
int[] V = new int[5];
int n=0;

int f (reference int x) {
    return x++; }

void foo(name int x, reference int y){
    x++; y++; x++; y++;}

V[0]=V[1]=V[2]=V[3]=V[4]=1;

foo(V[f(n)], V[f(n)]);
```

   Si dia lo stato del vettore `V` al termine dell'esecuzione del codice esposto (si ricordi che un comando della forma `return w++;` restituisce il valore corrente di `w` e poi incrementa `w` di uno).

5. Usando uno pseudolinguaggio che usi puntatori si fornisca un frammento di codice che generi un "dangling reference". Si faccia quindi vedere come con la tecnica delle "tombstones" non si ha più tale problema.

6. (Questo erauno vecchi fatto da Simone, che non abbiamo usato. Vero ?)

   Si assuma di avere uno pseudolinguaggio che adotti la tecnica del *reference count*; se *OGG* è un generico oggetto nello heap, indichiamo con `OGG.ref-c` il suo reference count (nascosto). Si consideri il seguente frammento di codice:

```
class C { int n; C next;}

C foo = new C();  // oggetto OG1 C bar = new C();  // oggetto OG2 foo.next = bar; bar = new C();
foo = bar;
```

Si dia il valore di OG1.ref-c, OG2.ref-c e OG3.ref-c dopo l'esecuzione del frammento. Quali di questi tre oggetti possono essere ritornati alla lista libera?

7. TROPPO FACILE ? Si forniscano in un qualsiasi pseudo linguaggio due tipi che siano equivalenti considerando l'equivalenza strutturale e che non lo siano considerando l'equivalenza per nome.

8. DA CAMBIARE Si consideri il seguente frammento di codice in un linguaggio nel quale il passaggio dei parametri avviene per nome.

```
{int x = 7;
int w = 1;

void fie(name int y,z){
     int x = 1;
     z = y + z + x;
     }

fie(x+w, w)
write(w); }
```

   Qual è il valore stampato da `write(w)` ?

9. DA CONTROLLARE Si considerino le seguenti classi Java:

```
public class A {
   int x = 4;
   int fie () {return x;}
   }

public class B extends A{
   int x = 6;
   int fie () { return x;}
   }
```

   Si consideri adesso il seguente frammento di codice:

```
B b = new B();
A a = b;
int zz = a.fie()+ a.x ;
```

   Si dica qual è il valore di `zz` al termine dell'esecuzione del frammento.

10. **Solo per il corso AL :**

    Se in un programma logico si cambia l'ordine degli atomi nel corpo di una clausola cambia la semantica del programma ? Motivare la risposta.

11. **Solo per il corso MZ a.a. 2003/04:** ?