Corso di Paradigmi di Programmazione
Prova scritta del 24 luglio 2012.

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

1. Si dia la definizione di linguaggio generato da una grammatica $G = (NT, T, S, P)$.

2. 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; se P è un programma in L_1 e x un suo dato, $\mathcal{I}^{L}_{L_1}(P, x)$ indica l'applicazione dell'interprete a P e x. Si dica se la valutazione di $\mathcal{I}^{L}_{L_1}(\mathcal{C}^{L_1}_{L_1,L_2}, \mathcal{C}^{L_1}_{L_1,L_2})$ produce un qualche risultato motivando la risposta.

3. Si consideri il seguente frammento di codice in uno pseudo linguaggio con scoping statico e passaggio dei parametri sia per valore che per nome.

```
int z=0;
int Omega(){
    return Omega();
}
int foo(int x, int y){
    if (x==0) return x;
    else return x+y;
}
write(foo(z, Omega()+z));
```

   (i) Si dica qual è il risultato dell'esecuzione di tale frammento nel caso in cui i parametri di `foo` siano passati *per nome*.

   (ii) Si dica qual è il risultato dell'esecuzione di tale frammento nel caso in cui i parametri di `foo` siano passati *per valore*.

4. Il linguaggio imperativo `L` è costituito dagli usuali comandi (assegnamenti, controllo di sequenza ecc.), ammette funzioni, ma, nel caso di funzioni ricorsive, queste devono essere ricorsive in coda. Inoltre permette comandi di allocazione (e deallocazione) esplicita della memoria. Si dica, motivando la risposta, qual è la più semplice forma di gestione della memoria utilizzabile nell'implementazione di `L`.

5. Si consideri la seguente definizione di tipo record:

```
type S = struct{
   int x;
   char y;
   };
```

   Si supponga che un `int` sia memorizzato su 2 byte, un char su 1 byte, su un'architettura a 16 bit con allineamento alla parola. In un blocco viene dichiarato un vettore:

```
S A[10];
```

   Indicando con `PRDA` il puntatore all'RdA di tale blocco, e con `ofst` l'offset tra il valore di `PRDA` e l'indirizzo iniziale di memorizzazione di `A`, si dia l'espressione per il calcolo dell'indirizzo dell'elemento `A[4].y`.

6. Si dica cosa viene stampato dal seguente frammento in un linguaggio con eccezioni:

```
{
void f() throws X {
   throw new X();
   }

void g (int sw) throws X {
   if (sw == 0) {f();}
   try {f();} catch (X e) {write("in␣g");}
   }
...
try {g(1);}
   catch (X e)  {write("in␣main");}
}
```

7. Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usa scope statico.

```
{
void f() {
        void g() {
        corpo_di_g;
        }

        void h() {
                void l(){
                corpo_di_l;
                }
        corpo_di_h;
        }
corpo_di_f;
}
```

   Si descriva graficamente l'evoluzione del display nella sequenza di chiamate f, g, h, l, g, supponendo che tutte le chiamate rimangano attive (ossia nessuna funzione ha restituito il controllo).

8. In un certo linguaggio di programmazione LP, un identificatore è una stringa su {a,b,0,1}, che inizia per a o b e può anche contenere, una sola volta, il carattere #. Si dia una grammatica il cui linguaggio generato è costituito da tutti e soli gli identificatori di LP .