Corso di Paradigmi di Programmazione
Prova scritta del 13 luglio 2009.

Tempo a disposizione: ore 2.

1. Si consideri la grammatica $G = (\{A, B\}, \{a, b\}, A, P)$ dove P è l'insieme seguente:

$$\begin{array}{lcl} A & ::= & aAa \mid B \\ B & ::= & BbbB \mid \epsilon \end{array}$$

   Si consideri la seguente affermazione: "Per ogni $k \geq 0$, $\mathcal{L}(G)$ contiene una stringa di lunghezza k". Si dica se è vera o falsa. Se si ritiene che sia vera, si fornisca una giustificazione (informale o una dimostrazione); se si ritiene falsa, si mostri un k ed una stringa lunga k che non sta in $\mathcal{L}(G)$.

2. Si consideri la seguente definizione di funzione in un linguaggio con gestione della memoria con pila di RdA:

```
int f(int n,m){
   if (n==0) return 1;
   else return f(n-1,m+n);
```

   Quanti record di attivazione sono certamente *necessari* su una macchina astratta per calcolare `f(3,0)`? Si dia una sintetica giustificazione.

3. In uno pseudolinguaggio con eccezioni (`try/catch`) si incontra il seguente blocco di codice:

```
public static void ecc() throws X {
   throw new X();
}
public static void g (int para) throws X {
   if (para == 0) {ecc();}
   try {ecc();} catch (X) {write(3);}
}
public static void main () {
   try {g(1);} catch (X) {write(1);}
   try {g(0);} catch (X) {write(0);}
}
```

   Si dica cosa viene stampato all'esecuzione di `main()`.

4. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per valore-risultato:

```
int X[10];
int i = 1;
X[0] = 10;
X[1] = 10;
X[2] = 10;
void foo (value-result int Y,J){
     X[J] = J-1;
     write(Y);
     J++;
     X[J]=J;
     write(Y);
     }
foo(X[i],i);
write(X[i]);
```

5. 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[3][3][4]` rispetto all'elemento `A[0][0][0]`? (Si risponda in notazione decimale).

6. Si assuma di avere uno pseudolinguaggio che adotti la tecnica dei *locks and keys*. Se *OGG* è un generico oggetto nello heap, indichiamo con `OGG.lock` il suo lock (nascosto); se *PTR* è un generico puntatore (sulla pila o nello heap), indichiamo con `PTR.key` la sua key (nascosta). Si consideri il seguente frammento di codice:

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

Si diano possibili valori di `OG1.lock, OG2.lock, foo.key, fie.key` e `bar.key` dopo l'esecuzione del frammento.

7. È dato il seguente frammento di codice in uno pseudolinguaggio con scope statico gestito con display:

```
int x = 5;
int y = 4;
void B(){
   int x = 4;
   int z = 3;
   C();
}
void C(){
   int x = 3;
   void D(){
      int x = 2;
   }
   D();
}
B();
```

Si rappresenti graficamente il display subito dopo che il controllo è entrato nella funzione `D`.

8. Si considerino le seguenti classi Java:

```
public class A {
   int x = 5;
   int fie () {return g();}
   int g() {return x;}
   }

public class B extends A{
   int x = 0;
   int g() {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.