Corso di Paradigmi di Programmazione
Prova scritta del 5 luglio 2011.

Tempo a disposizione: ore 2.

SCRIVERE LE SOLUZIONI A 1-4 E 5-8 SU DUE FOGLI DIVERSI

1. (i) Le seguenti due grammatiche sono entrambe libere da contesto? (ii) Quale delle due genera il linguaggio $L = \{a^n b^m a^m b^p \mid m, n, p \geq 0\}$? Per l'altra grammatica, si dia una stringa generata dalla grammatica e non appartenente a L:

$$(1)\quad \begin{array}{l} S \to AbSaB \mid \varepsilon \\ A \to aA \mid \varepsilon \\ B \to aB \mid \varepsilon \end{array} \qquad (2)\quad \begin{array}{l} S \to ACB \mid \varepsilon \\ A \to aA \mid \varepsilon \\ B \to aB \mid \varepsilon \\ C \to aCb \mid \varepsilon \end{array}$$

2. Si consideri la seguente frase, tratta dal manuale di Java: *In un programma corretto, ogni variabile deve aver avuto assegnato un valore prima di poter essere utilizzata nella parte destra di un assegnamento.* Si dia, se possibile, una grammatica che esprime formalmente questo vincolo. Se non è possibile, si motivi brevemente.

3. 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):

$$A \to , \quad aa \to A, \quad AB \to aA, \quad A \to A, \quad B \to aA$$

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

```
int x = 2;
void foo(value-result int y){
    x = x+1;
    y = y+10;
    x = x+y;
    write(x);
}
{int x = 10;
 foo(x);
 write(x);
}
```

5. Cosa è una *dangling reference*? È possibile crearne una in Java? Motivare brevemente.

6. Si consideri la seguente dichiarazione di array multidimensionale

```
int x;
read(x);
int A[10][x];
```

   dove il comando `read(x)` permette di leggere il valore della variabile `x` dall'esterno.

   Come viene memorizzato l'array A ? Inoltre 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.); il valore di x letto e' 3. Qual è l'offset dell'elemento `A[1][5]` rispetto all'inizio dell'array? (Si risponda in notazione decimale).

7. Si considerino le seguenti dichiarazioni in Java:

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

   Nello scope di tali dichiarazioni, qual è il valore dell'espressione `a.f(1)` ?

8. In un certo linguaggio, il tipo T è compatibile col tipo S. Nello scope delle dichiarazioni:

```
S s;
T t;
int g(S x){...}
```

   si consideri l'espressione

```
g(t) + g(s)
```

   La funzione `g` è necessariamente *overloaded*? La macchina astratta inserisce in tale espressione coercizioni? In caso positivo, dove?