Corso di Paradigmi di Programmazione
Prova scritta del 10 giugno 2009.

Tempo a disposizione: ore 2.

1. Si dica se la seguente grammatica è ambigua o meno, motivando quanto asserito:

$$(\{S, C, B\}, \{\texttt{if}, \texttt{then}, \texttt{else}, \texttt{skip}, \texttt{true}, \texttt{false}, \texttt{endif}, ;\}, S, R)$$

   dove R è il seguente insieme di produzioni:

$$\begin{array}{l} S \rightarrow C; \\ C \rightarrow \texttt{skip} \mid \texttt{if B then C else C endif} \mid \texttt{if B then C endif} \\ B \rightarrow \texttt{true} \mid \texttt{false} \end{array}$$

2. Alcuni linguaggi di programmazione, usano per il comando `for` una sintassi diversa a seconda del fatto che l'incremento dell'indice sia positivo o negativo. Come mai? E come mai altri linguaggi usano la stessa sintassi per entambi i casi?

3. Si consideri un linguaggio con scope statico, implementato mediante catena statica, nel quale tutti gli identificatori sono noti staticamente. Si assuma inoltre che ogni nome usato sia dichiarato in un solo blocco. Si prendano in considerazione le operazioni di (NL)="accesso ad una variabile non locale x" e (L)="accesso ad una variabile locale y" (nel contesto di un blocco). Per ognuna delle due operazioni di dica a quale dei seguenti parametri è proporzionale il tempo necessario all'esecuzione dell'operazione, motivando brevemente la risposta.

   (i) Il numero di variabili presenti nel programma; (ii) Il numero di variabili presenti nei blocchi compresi tra quello di dichiarazione della variabile e quello in cui si accede alla variabile stessa; (iii) Il numero di record di attivazione presenti sulla pila e compresi tra quello contenente la dichiarazione della variabile e quello in cui vi si accede; (iv) il numero di blocchi che contengono testualmente il blocco in cui si accede alla variabile e che sono contenuti in quello nel quale la variabile è dichiarata; (v) Il tempo è costante, e quindi indipendente da questi parametri; (vi) nessuna delle risposte precedenti.

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

```
int X[10];
int i = 1;
X[0] = 10;
X[1] = 10;
X[2] = 10;
void foo (reference 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 il seguente frammento in un linguaggio con garbage collector mediante contatori dei riferimenti (le variabili adottano un modello a riferimento):

```
class Elem { int n; C next;}
Elem foo(){
   Elem p = new C();      // oggetto OGG1
   p.next = new C();  // oggetto OGG2
   Elem q = new C();      // oggetto OGG3
   q.next = p.next;
   return p.next;
   }
Elem r = foo();
```

   Si dica quali sono i valori dei contatori dei riferimenti dei tre oggetti al termine dell'esecuzione del frammento.

6. Si consideri un linguaggio con scope statico per tutti i nomi, nel quale i nomi delle eccezioni devono essere dichiarati con la sintassi

   `exception` *`nomeeccezione`*`;`

   Qual è l'effetto dell'esecuzione del seguente frammento?

```
exception X;

void f(){
    throw X;
 }

void B(){
    exception X;
    void g(){
        try {f();} catch (X){
                    print("in␣g");
                    }
    }
}

B();
```

7. 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[1][2][4]` rispetto all'elemento `A[0][0][0]`? (Si risponda in notazione decimale).

8. Un programma Java dichiara alcune classi e alcune interfacce. Relativamente alla relazione di sottotipo *immediata* (cioè che non prende in considerazione l'eventuale transitività della relazione), si sa che: `A` è un sottotipo di `B` e di `E`; `B` è un sottotipo di `C` e `D`; `E` è un sottotipo di `D` e `F`. Inoltre, si sa che `F` è una classe. Quali degli altri tipi corrispondono certamente ad interfacce?