

Tempo a disposizione: ore 2.

1. Si spieghi sinteticamente cosa è un blocco e a cosa serve.
2. Si consideri la seguente grammatica $G = (\{S, T\}, \{a, b\}, S, P)$ dove P è dato dalle seguenti produzioni:

$$\begin{aligned} S &::= SaS \mid TbT \mid T \\ T &::= TaT \mid S \mid a \end{aligned}$$

Si mostri che G è ambigua.

3. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico 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;
}

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

write(x); }
```

4. 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. 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.

6. Si dica a cosa servono le regole di equivalenza di tipo in un linguaggio di programmazione. Si descrivano poi sinteticamente i due approcci più comuni all'equivalenza.
7. **Solo per: corso AL; corso MZ a.a. 2002/03** 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)` ?

8. **Solo per il corso MZ:** Si considerino le seguenti classi Java:

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

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

Si consideri adesso il seguente frammento di codice:

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

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

9. **Solo per il corso AL: ***DA CAMBIARE***** Si consideri il seguente programma logico

```
p(X):- q(a), r(Y).
q(b).
q(X):- p(X).
r(b).
```

Si dica se il goal `p(b)` termina o meno, giustificando la risposta (ricordiamo che `X,Y` sono variabili e `a,b` costanti).

10. **Solo per il corso MZ a.a. 2003/04:** Si elenchino le tre principali modalità di valutazione utilizzate per i linguaggi funzionali. Quale di esse è quella utilizzata da Scheme?