

Tempo a disposizione: ore 2.

1. Si fornisca una grammatica libera da contesto ambigua.
2. Cosa stampa il seguente frammento di codice in un linguaggio con scope statico e passaggio per valore-risultato, e che valuta le sue espressioni da sinistra a destra?

```
int x = 1;
int A[5];
int foo () {return x++;}
for (int i=0, i<5, i++)
  A[i]=i;
int f(value-result int a){
  int x = 5;
  a = x+a;
  a = x+a;
  return a;
}
write (f(A[foo()+x]) + A[x] + x);
```

3. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico e deep binding:

```
int x = 3;
procedure ass_x(n:int)
  {x = n;
  }
procedure stampa_x
  {write_integer(x);
  }
procedure pippo(function S, P ; int n )
  { int x= 10;
    if n=1 then ass_x(n)
      else S(n);
    P;
    stampa_x
  }

pippo(ass_x, stampa_x, 1);
pippo(ass_x, stampa_x, 2);
```

4. Usando uno pseudolinguaggio che usi puntatori si fornisca un frammento di codice che generi un “dangling reference”. Si faccia quindi vedere come con la tecnica dei “locks and keys” non si ha più tale problema.
5. Si dica cosa è l’aliasing e perché può rendere difficoltosa la verifica della correttezza dei programmi. E’ possibile inserire nel compilatore di un ipotetico linguaggio un controllo che permetta di identificare tutte le situazioni di aliasing ? Perché ?
6. Si forniscano in un qualsiasi pseudo linguaggio: (i) due tipi che siano equivalenti considerando l’equivalenza strutturale e che non lo siano considerando l’equivalenza per nome; (ii) due tipi che sono compatibili ma non equivalenti.
7. Si considerino le seguenti dichiarazioni in un generico linguaggio orientato a oggetti

```
class A{
  int x = 5;
  int f(int n){return n+1;}
}
class B extending A{
  int x = 2;
  int y = 2;
  int g(){return f(3);}
}
```

Si descrivano almeno due modi possibili per implementare le classi e realizzare il meccanismo di ereditarietà. Supponendo poi che la classe A sia modificata come segue

```
class A{
    int x = 5;
    int h(){return 3;}
    int f(int n){return n+1;}
}
```

quale delle soluzioni proposte permette di gestire meglio tale modifica ?

8. **Solo corso AL** Supponiamo di rappresentare i naturali usando 0 per lo zero e $s(N)$ per il successore di N . Si dica che cosa producono le valutazioni dei goal $p(s(s(0)), s(0), X)$ e $p(s(0), X, p(s(s(0))))$ nel seguente programma.

```
p(0, X, X).
p(s(Y), X, s(Z)):- p(Y, X, Z).
```

9. **Solo corso MZ**

* TO DO **