

Tempo a disposizione: ore 2.

1. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope dinamico

```
int x = 10;
void foo( int y, int z){
    x = x-1;
    y = y+10;
    z = z+y;
    write(x);
}
{int x = 50;
  foo(x, x);
  write(x);
}
```

2. L'esecuzione del seguente frammento di codice su una certa implementazione risulta nella stampa del valore 2.

```
int x = 10;
void foo( int y, int z){
    x=0;
    y=z+z;
}
int V[10];
for (int i=0, i<10, i++) V[i]=i;

{ int x = 4;
  foo(V[x],V[x++])
  write (V[x]);
}
```

Si fornisca una possibile spiegazione.

3. Si consideri il seguente frammento in uno pseudolinguaggio con parametri di ordine superiore:

```
{int m = 15;
 int n = 15;
 void foo (int f(), int n){
     int m = 25;
     if (n==0) {int n = 35;
                f();
            }
     else      {m = 45;
                n = 45;
                foo(f,0);
            }
 }
 int g(){
     write(n,m);
 }

{int m = 55;
  foo(g,1);
}
}
```

Si dica cosa stampa il frammento con con scope statico e deep binding.

4. Si considerino le seguenti definizioni di classe in Java:

```
class A{
    int x;
    int f (int y){return y+1;}
    int g (int k){return k+2;}
}
class B extends A{
    int x;
    int y;
    void g (int z){return y;}
}
class C extends B{
    int f (int y){return y+3;}
    int h (int y){return y+3;}
}
```

Si supponga che la gerarchia delle classi sia implementata mediante vtable. Qual è la struttura della vtable di C? Se nella classe A eliminiamo la definizione del metodo f cambia la vtable di C? Motivare la risposta.

5. Si consideri la seguente definizione di tipo record:

```
type S = struct{
    float x;
    char y;
};
```

Si supponga che un float sia memorizzato su 4 byte, un char su 1 byte, su un'architettura a 32 bit con allineamento alla parola. In un blocco viene dichiarato un array:

```
S A[10,10];
```

Indicando con PRDA il puntatore all'RdA di tale blocco, e con *ofst* l'offset tra il valore di PRDA e l'indirizzo iniziale di memorizzazione di A, supponendo che l'array sia memorizzato per riga si dia l'espressione per il calcolo dell'indirizzo dell'elemento A[3,4].y.

6. Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usa scope statico.

```
{
void f() {
    void g() {
        corpo_di_g;
    }

    void h() {
        void l(){
            corpo_di_l;
        }
        corpo_di_h;
    }
    corpo_di_f;
}
```

Si descriva graficamente l'evoluzione del display nella sequenza di chiamate f, h, l, g, h supponendo che tutte le chiamate rimangano attive (ossia nessuna funzione ha restituito il controllo).

7. Usando uno pseudolinguaggio che ammetta l'uso dei 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. Avremmo potuto usare la tecnica del contatore dei riferimenti invece del "locks and keys" ? Motivare la risposta.
8. Si descriva brevemente il funzionamento di una Remote Procedure Call (RPC) illustrando anche la differenza rispetto al rendez-vous.