

Tempo a disposizione: ore 2.

1. Si inserisca del codice al posto degli asterischi e si indichi la modalità di passaggio dei parametri da usarsi per far sì che il programma seguente stampi 2, 3 e 4. Devono essere rispettate le seguenti restrizioni: non si possono dichiarare nuove variabili; la valutazione del parametro attuale al momento della chiamata di foo non può restituire 3; i comandi `***` nel corpo della procedura foo non possono modificare la variabile X.

```
inx X = 2;
void foo ( int Y ){
***;
write(X);
write(Y);
***;
}
foo(***);
write(X);
```

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

```
{int x = 10;
int h(){
    write(x);
}
void foo (int f(), int n){
    int x = 30;
    int g(){
        write(x);
    }
    if (n==0)    {f();
                  g();
                }
    else        {x = 40;
                  foo(g,0);
                }
}
{int x = 20;
foo(h,1);
}
}
```

Si dica cosa stampa il frammento con con scope dinamico e shallow binding.

3. 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).

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

```
class A{
    int x = 10;
    int f (int y){return g(y);}
    int g (int k){return -k ;}
}
class B extends A{
    int x;
    int y;
    int g (int z){return z;}
    int f (int y){return 1;}
    int p (int z){return g(y);}
}
class C extends B{
    int x = 100;
    int g (int k){return k*x;}
    int h (int y){return y+3;}
}
C ogg = new C;
A pippo = ogg;
int w = pippo.f(3)+ pippo.x
```

Si supponga che la gerarchia delle classi sia implementata mediante vtable. Si mostri la rappresentazione dell'implementazione dell'oggetto `ogg`, e delle vtable di A, B e C. Si dica che valore viene assegnato a `w` motivando la risposta.

5. Si considerino le seguenti dichiarazioni (Pascal):

```
type stringa = packed array [1..16] of char;
type punt_stringa = ^stringa;
type persona = record
    nome = stringa;
    case studente: Boolean of
        true: (matricola: integer);
        false: (codicefiscale: punt_stringa)
end;
```

e si supponga che la variabile `C` contiene il puntatore alla stringa "CODICEPIPPO". Si descriva il layout di memoria dopo ognuna delle seguenti istruzioni:

```
...
var pippo persona;

pippo.studente:= true;
pippo.matricola := 223344;
pippo.studente:= true;
pippo.codicefiscale := C;
```

6. E' dato il seguente programma Prolog (X e Y sono variabili mentre a e b sono costanti) .

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

Si dica se il goal `?p(a)` termina o meno, giustificando la risposta.

7. Ricordiamo che Scala utilizza il modello computazionale per sostituzione. Sia `loop` definito come segue

```
def loop: Int = loop
```

e si consideri la funzione `first` per la proiezione a sinistra:

```
def first(x: Int, y: Int) = x
```

- (a) Descrivere la valutazione dell'espressione `first(5, loop)` se si utilizza una strategia di valutazione per valore (*call-by-value*).
- (b) Fare lo stesso rispetto alla strategia di valutazione per nome (*call-by-name*).
8. E' possibile simulare la comunicazione asincrona mediante quella sincrona? E, viceversa, simulare la comunicazione sincrona mediante quella asincrona? Motivare brevemente le risposte.