

Tempo a disposizione: ore 2.

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

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

2. Il linguaggio imperativo Ric è costituito dagli usuali comandi (assegnamenti, controllo di sequenza ecc.), permette comandi di allocazione e deallocazione esplicita della memoria, ammette funzioni, ma, nel caso di funzioni ricorsive, queste devono essere ricorsive in coda. Si dica, motivando la risposta, qual è la più semplice forma di gestione della memoria utilizzabile nell'implementazione di Ric.
3. Si consideri il seguente frammento di codice scritto nello pseudolinguaggio Cfinto

```
struct Nodo{
    int key; // chiave intera
    struct Nodo* next; // puntatore al prossimo nodo
};
typedef Nodo* Lista;
Lista pippo;
pippo = crealista();

while (pippo != NULL) && (*pippo.Key != 4) do
    pippo = *pippo.next;
```

(dove `crealista` e' una funzione che restituisce il puntatore ad una lista). Con una implementazione di Cfinto l'esecuzione del precedente frammento produce un errore mentre con un'altra implementazione l'esecuzione termina senza errori. Si fornisca una possibile spiegazione, sapendo che entrambe le implementazioni sono corrette.

4. 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) {f();
                g();
            }
     else      {m = 45;
                n = 45;
                foo(f,0);
            }
 }
 int g(){
     write(n,m);
 }

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

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

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

```
class A{
    int x;
    int f (int y){return g(y);}
    int g (int k){return k+2;}
}
class B extends A{
    int x;
    int y;
    int g (int z){return z;}
}
class C extends B{
    int z;
    int h (int y){return y+3;}
}
C ogg = new C;
int w = ogg.f(3);
```

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.

6. Si consideri la seguente definizione di tipo record:

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

Si supponga che un `integer` sia memorizzato su 2 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[4,4];
```

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[2,1].y`.

7. 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 della catena statica nella sequenza di chiamate `f`, `h`, `l`, `g`, `h` supponendo che tutte le chiamate rimangano attive (ossia nessuna funzione ha restituito il controllo).

8. Si descrivano brevemente analogie e differenze fra semafori e monitor.