

Tempo a disposizione: ore 2.

**Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.**

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

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

2. Si dica cosa stampa il seguente frammento di programma assumendo scope statico e deep binding:

```
{int x = 10;
int n = 100;
int stampa(){
    write(x);
}
void foo (int f(), int g()){
    int x = 30;
    int n = 300;
    stampa()
    f();
    g();
}
{int x = 20;
int n = 200;
int ass_x(){
    x= x+n;
}
int stampa(){
    write(x - 1);
}
foo(ass_x(), stampa());
}
```

3. Si consideri la seguente definizione di tipo record:

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

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 Arr [8,8];
```

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 Arr, supponendo che l'array sia memorizzato per riga si fornisca l'espressione per il calcolo dell'indirizzo dell'elemento Arr[3,4].x.

4. 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 "tombstones" non si ha più tale problema.