

# Esame di LP1

2 Marzo 2017

## Domande generali – Max 8 punti

**Esercizio 1: [2 punti]** Barrare tutte le frasi vere.

1. Nel primo Fortran l'occupazione di memoria di un programma era nota a *compile time*.
2. HTTP (senza scripts) è un linguaggio *general purpose*.

**Esercizio 2: [2 punti]** In C e C++, le parti sinistra e destra dell'assegnamento  $x[1] = *(y+2);$  denotano rispettivamente:

- a)  $mem(env(x)+1)$  e  $mem(mem(env(y)+2))$
- b)  $mem(x+1)$  e  $mem(mem(env(y))) + 2$
- c)  $env(x)+1$  e  $mem(env(y)) + 2$
- d)  $env(x)+1$  e  $mem(mem(env(y))+2)$

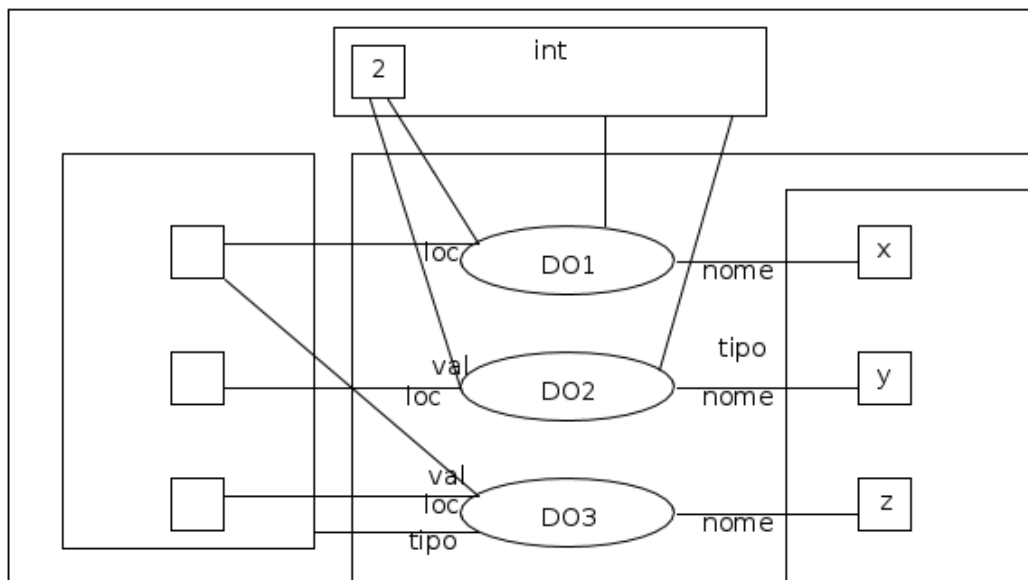
**Esercizio 3: [2 punti] (Barrare tutte le risposte corrette)** Date le dichiarazioni in C:

```
typedef float Dollars;  
typedef double Euros;  
Dollars d = 10.0;  
Euros e;
```

Dire se l'assegnamento  $e = d;$

rispetta la name equivalence     rispetta la structural equivalence     è corretto in C

**Esercizio 4: [2 punti]** Scrivere nel riquadro un codice C che generi i seguenti data object:



## Esercizio sul passaggio di parametri – Max 11 punti

Dire qual è l'output del seguente programma nei casi elencati qui sotto:

1. Scoping dinamico, [MODE] = IN per copia
2. Scoping dinamico, [MODE] = IN per riferimento
3. Scoping statico, [MODE] = IN per copia
4. Scoping statico, [MODE] = IN OUT per riferimento

Mostrare gli stack di attivazione (pena la perdita di punti), tranne nei casi di errore, nei quali bisogna invece indicare l'istruzione che causa l'errore.

```
program p1
int x; int y; int z; int t;
  procedure p2([IN x copia] int y,[IN x copia] int x)
    int z;
    BEGIN
    z=x;
    x=4;
    t=z-1;
    write(x,y,z,t);
    END

  procedure p3([MODE] int x,[IN x copia] int t)
    int z;
    procedure p4([IN x copia] int x)
      int t;
      BEGIN
      t=x-2;
      z=4;
      y=y;
      p2(x, t);
      write(x,y,z,t);
      END

    BEGIN
    z=t;
    x=z*2;
    t=x;
    y=z*1;
    p4(y);
    write(x,y,z,t);
    END

BEGIN
x=1;
y=1;
z=1;
t=2;
p3(z, y);
write(x,y,z,t);
END
```

## UML – Max 13 punti

Si vuole modellare una rete ferroviaria. I nodi della rete possono essere solamente stazioni o scambi. Tutti i nodi hanno un nome. Inoltre le stazioni hanno un indirizzo mentre gli scambi hanno una coppia di coordinate (latitudine, longitudine). Gli archi sono etichettati con un numero che rappresenta la lunghezza del tratto ferroviario corrispondente. I percorsi sono liste concatenate di nodi della rete. Uno stesso nodo della rete ferroviaria può appartenere a più percorsi. Ai percorsi si può appendere un nuovo nodo.

**Esercizio 1:** Disegnare un diagramma delle classi per queste specifiche.  
**[max 7 punti]**

**Esercizio 2:** Disegnare un diagramma delle classi che mostra come le classi di associazione vengono trasformate in classi semplici per poter essere implementate.  
**[max 2 punti]**

**Esercizio 3:** Disegnare un diagramma delle sequenze per lo scenario seguente, mostrando tutte le interazioni tra gli oggetti coinvolti. **Se un oggetto ha bisogno di un attributo privato di un altro oggetto dovete mostrare come se lo procura.**

Un utente aggiunge il nodo S1 della rete ferroviaria in coda ad un percorso che già contiene il nodo S2.

**[max 4 punti]**