

Esame di LP1

7 Febbraio 2017

Domande generali – Max 8 punti

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

1. Il C++ ha un garbage collector.
2. Il primo Fortran supportava una operazione analoga a una malloc.

Esercizio 2: [2 punti] In C e C++, l'assegnazione le cui parti sinistra e destra denotano $env(x)+1$ e $mem(mem(env(y))+1)$, rispettivamente, è:

- a) `x[1] = *(y+1);`
- b) `*(x+1) = y+1;`
- c) `x[1] = y+1;`
- d) `x[1] = y[1];`

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

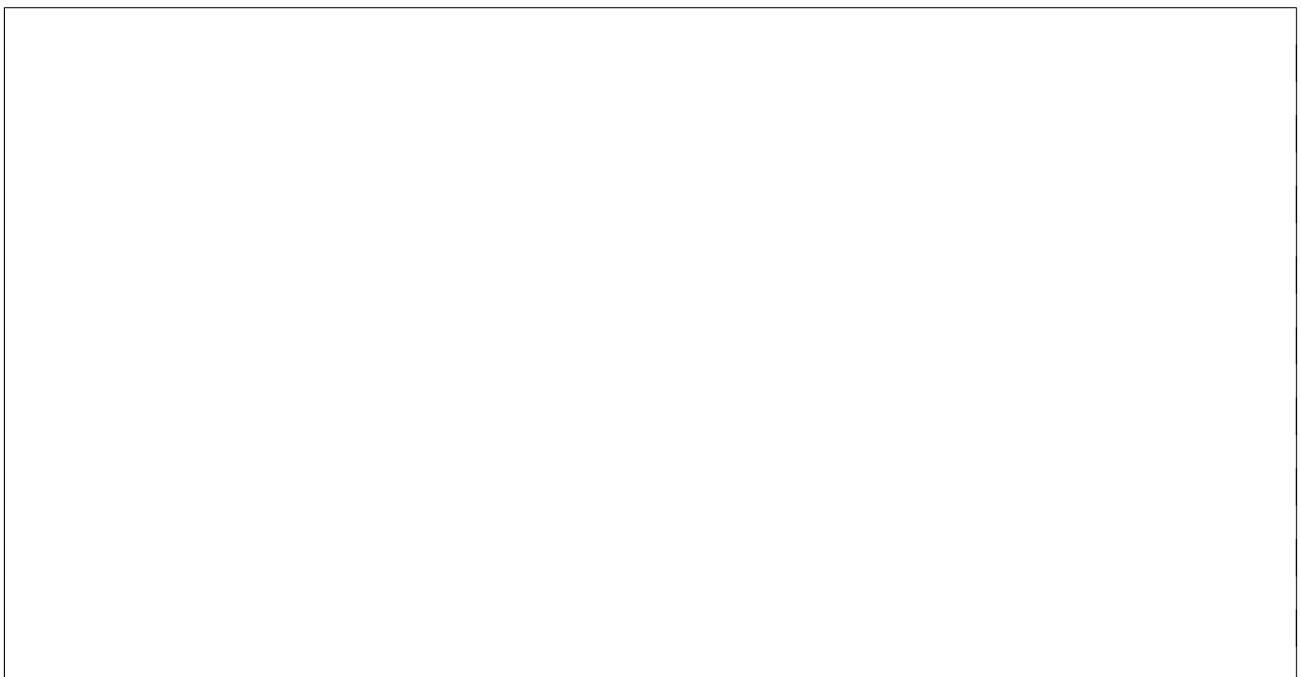
```
typedef float Dollars;  
typedef float Euros;  
Dollars d = 10.0;  
Euros e;
```

Dire se l'assegnamento `e = d;`

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

Esercizio 4: [2 punti] Disegnare i data object generati da questo codice C:

```
int x[2], *y;  
x[0]=1; y=x+1;
```



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 statico, [MODE] = IN per copia
3. Scoping statico, [MODE] = IN OUT 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 OUT x rif] int z)
    int y;
    BEGIN
    y=3;
    z=4;
    t=y-1;
    if x=3 then x=t else x=y;
    p3(x);
    write(x,y,z,t);
    END

    procedure p3([MODE] int t)
    int y;
    BEGIN
    y=t;
    t=2;
    z=y;
    if x<1 then x=t+3 else x=1;
    p4(y);
    write(x,y,z,t);
    END

    procedure p4([IN x copia] int t)
    int z; int x;
    BEGIN
    z=3;
    x=y+2;
    t=4;
    y=3;
    write(x,y,z,t);
    END

BEGIN
x=0;
y=0;
z=2;
t=1;
p2(t);
write(x,y,z,t);
END
```

UML – Max 13 punti

Si vuole progettare una applicazione gestionale per il mobilificio IDEA. Ogni negozio IDEA ha un indirizzo e una lista concatenata di ordini. Ogni ordine ha un numero identificativo e un articolo. Gli articoli hanno una stringa identificativa e possono essere di soli due tipi: interni ed esterni. Quelli interni contengono la fabbrica IDEA di provenienza, descritta con una stringa. Quelli esterni hanno un fornitore, caratterizzato da nome e indirizzo. Ogni negozio IDEA fa uso di un database di backend progettato da altri, di cui si sa unicamente che supporta il metodo *query*(String *q*):String. Dai negozi si può eliminare il primo ordine della lista.

Esercizio 1: Disegnare un diagramma delle classi per queste specifiche.
[max 8 punti]

Esercizio 2: 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 rimuove da un negozio N1 il primo ordine O1. La nuova testa della lista di ordini diventa il suo successore O2. L'oggetto O1 viene distrutto.

[max 6 punti]