

Esame di LP1

24 Luglio 2017

Domande generali – Max 8 punti

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

1. Nel paradigma funzionale si possono usare cicli *for* e *while*.
2. Nel paradigma logico non si distinguono parametri di input e output.

Esercizio 2: [2 punti] In C e C++, le parti sinistra e destra dell'assegnamento $x[1] = y[0]$; denotano rispettivamente:

- a) $mem(env(x))+1$ e $mem(env(y))+0$
- b) $mem(env(x))+1$ e $mem(env(y) + 0)$
- c) $mem(env(x)+1)$ e $mem(env(y)+0)$
- d) $env(x)+1$ e $mem(env(y))$

Esercizio 3: [2 punti] (Barrare tutte le risposte corrette) Scrivere nel riquadro due *typedef* e un assegnamento tali che l'assegnamento: (i) rispetti l'equivalenza strutturale; (ii) *non* rispetti la name equivalence; (iii) *non* sia corretto in C.

Esercizio 4: [2 punti] Disegnare nel riquadro i data objects prodotti dalle seguenti istruzioni:

```
int *p = (int*) malloc(sizeof(int));  
*p = 2;
```

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 a; int b; int c; int d;
  procedure p2([IN OUT x rif] int b)
  int c;
  BEGIN
  c=b;
  b=1;
  a=b;
  d=3;
  write(a,b,c,d);
  END

  procedure p3([IN OUT x rif] int c,[MODE] int a)
  int b;
    procedure p4([IN OUT x rif] int d,[IN x copia] int b)
    int c;
    BEGIN
    c=2;
    if d>8 then d=b*2 else d=1;
    b=1;
    if c<9 then a=4 else a=d*2;
    p2(c);
    write(a,b,c,d);
    END

  BEGIN
  b=3;
  c=2;
  a=3;
  d=1;
  p4(d, a);
  write(a,b,c,d);
  END

BEGIN
a=1;
b=4;
c=2;
d=1;
p3(a, d);
write(a,b,c,d);
END
```

UML – Max 13 punti

Si vuole progettare un sistema di gestione per la piscina del CUS. La piscina ha degli utenti, a cui può accedere usando il nome come chiave. Oltre al nome, ad ogni utente sono associate la data di nascita e una lista concatenata di ingressi (caratterizzati da data e ora). Agli utenti si può aggiungere un ingresso, che verrà posto in coda alla lista dell'utente. Gli utenti sono esclusivamente di due tipi: universitari ed esterni. agli universitari è associata la scansione di un documento che prova di essere docenti o studenti. L'implementazione delle scansioni non è specificata (potrebbero essere bitmap, PNG, PDF o altro); si sa solo che le scansioni possono essere visualizzate e stampate con i metodi *display()* e *print()*, che restituiscono void.

Esercizio 1: Disegnare un diagramma delle classi per queste specifiche.
[max 7 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 amministratore aggiunge l'ingresso con data D1 e ora O1 a un utente che nella sua lista ha già due ingressi precedenti. L'inserimento in coda alla lista è *ricorsivo*.

[max 6 punti]