

# Esame di LP1

Prof Piero Bonatti

18 Marzo 2016

## Domande generali – Max 8 punti

**Esercizio 1: [4 punti]** Barrare tutte le frasi vere.

1. La proprietà di invertibilità dei linguaggi logici prende il nome dal fatto che un unico programma può essere usato per calcolare sia una funzione sia la sua inversa [ ]
2. Il primo Fortran aveva uno stack di attivazione [ ]
3. Il linguaggio SQL è computazionalmente completo [ ]
4. In C++ gli interi (`int`) sono lunghi 32 bit [ ]

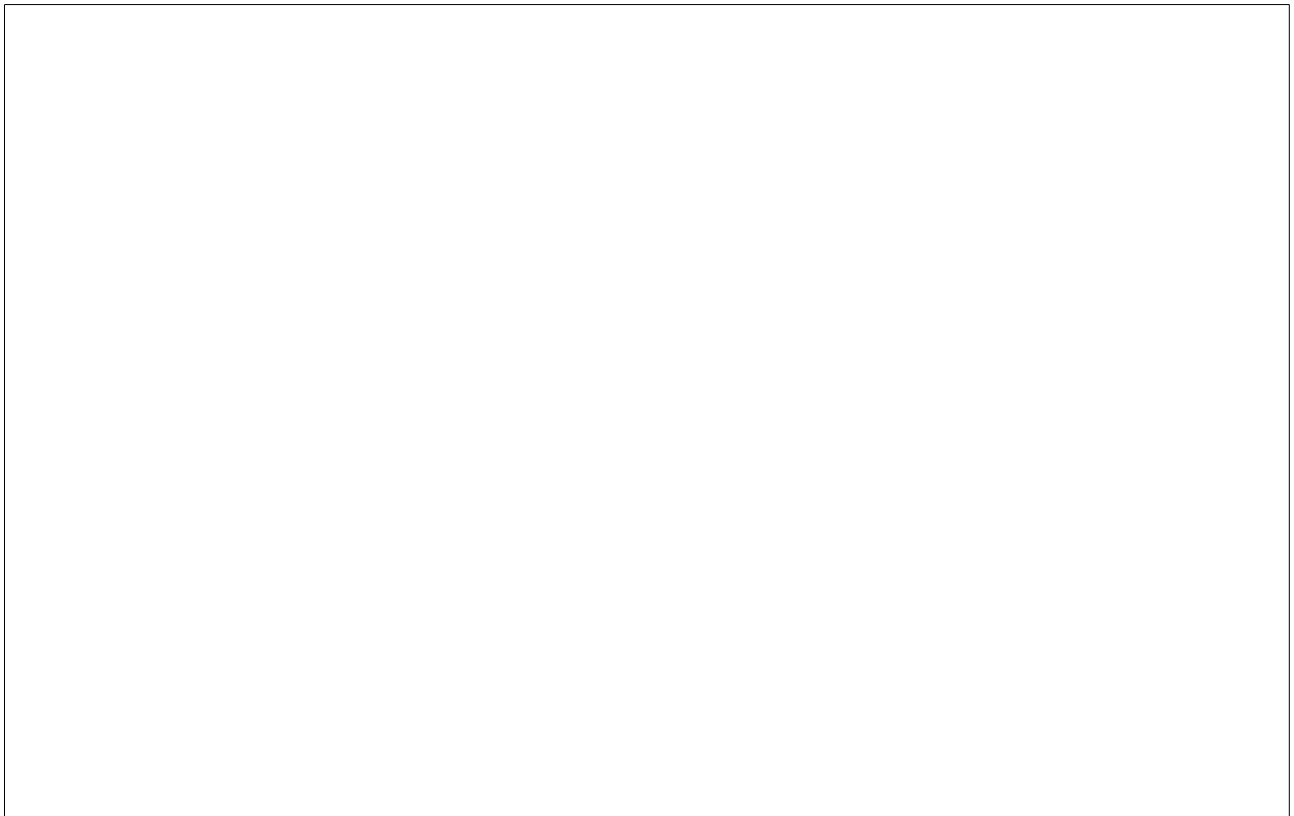
**Esercizio 2: [2 punti]** Date le dichiarazioni

```
int x, *y, v[10];
```

e sapendo come gli identificatori corrispondono a *mem* e *env*, dire quali degli assegnamenti qui sotto sono corretti in C e C++:

- a) `&x = y;` [ ]
- b) `v = &x;` [ ]
- c) `y = y+1;` [ ]
- d) `v = v+1;` [ ]

**Esercizio 3: [2 punti]** Disegnare il diagramma che rappresenta l'implementazione puramente compilata più generale (che comprende i casi di cross-compilation)



## 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] = OUT per riferimento
3. Scoping statico, [MODE] = IN per copia
4. Scoping statico, [MODE] = IN OUT per copia

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 q; int r; int s; int t;
  procedure p2([IN x copia] int s,[IN x copia] int q)
    int r;
      procedure p3([IN OUT x rif] int r)
        int q;
        BEGIN
          q=r*3;
          r=r+2;
          s=2;
          t=q;
          p4(t, s);
          write(q,r,s,t);
        END

      BEGIN
        if s<5 then r=q*2 else r=4;
        s=r;
        q=1;
        t=3;
        p3(t);
        write(q,r,s,t);
      END

    procedure p4([MODE] int q,[IN x copia] int t)
      int s;
      BEGIN
        s=4;
        q=2;
        t=s+4;
        r=t;
        write(q,r,s,t);
      END

  BEGIN
    q=4;
    r=3;
    s=0;
    t=1;
    p2(r, q);
    write(q,r,s,t);
  END
```

## UML – Max 13 punti

Si vuole descrivere il gestore di processi di un sistema operativo. Esso dispone di un intero che rappresenta l'intervallo di tempo massimo passato il quale avviene il cambio di contesto (cioè si sospende un processo e si attiva il successivo).

Il gestore di processi mantiene i descrittori di processo in una lista concatenata circolare. I descrittori di processo hanno un numero di processo, un vettore di  $n$  registri in cui salvare lo stato al momento del cambio di contesto, e sono associati all'insieme delle proprie pagine di memoria virtuale (caratterizzate da un numero e dall'indirizzo fisico della memoria in cui si trovano), alle quali accedono per chiave, utilizzando il numero di pagina. I descrittori di processo permettono di salvare al proprio interno lo stato dei registri della CPU e di ripristinarlo.

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

Il gestore chiede al descrittore del processo in esecuzione di salvare lo stato dei registri, poi passa al descrittore successivo nella lista circolare e gli chiede di ripristinare i propri registri.

**[max 6 punti]**