

Politecnico di Milano
Facoltà di Ingegneria dell'Informazione
Informatica 3
Prof. Giovanni Agosta
Prof. Pier Luca Lanzi
Prof. Marcello Restelli

COGNOME E NOME (IN STAMPATELLO)

MATRICOLA

Appello del 14 Febbraio 2008

Risolvere i seguenti esercizi, scrivendo le
risposte ed eventuali tracce di soluzione negli
spazi disponibili.

Non consegnare altri fogli.

Spazio riservato ai docenti

--	--	--	--	--

Esercizio 1. Si consideri il segmento di codice scritto in un linguaggio con scope statico ispirato al C.

- (1) si disegni il contenuto dello stack appena prima dell'esecuzione dell'istruzione "@@" durante l'esecuzione di "P2(A, 20, &B);"
- (2) dove necessario, si calcolino le coppie <distanza, offset> delle variabili.
- (3) si esprima in SIMPLESEM l'assegnamento etichettato con @@

```
#include <stdio.h>

typedef struct {double x,y;} point;

void DrawRectangle(point UC, point LC) { };

void P1(point A)
{
    void P2(point A, double d, point *B)
    {
        B->x = A.x+d;
        B->y = A.y+d;      /* @@ */
    }

    point B;
    P2(A, 10, &B);
    DrawRectangle(A,B);
    P2(A, 20, &B);
    DrawRectangle(A,B);
}

main()
{
    double x1,y1;
    point A;

    scanf("%g %g", &x1, &y1);
    A.x = x1;
    A.y = y1;
    P1(A);
}
```

Esercizio 1 (continua).

	Contenuto	Commento
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		

Esercizio 2. Dato il seguente codice Python: calcolare l'output del programma, per ogni istruzione di output giustificare il valore delle variabili stampate (output non giustificati non verranno presi in considerazione).

```
def a(b, c) :  
    if b :  
        c+=b*2  
    else :  
        c=c*2
```

```
b=[[ ]]  
c=[b, 1]  
a(b, c)  
print c  
a(a(b, c), c)  
print c  
print a(0, c)+b
```

Esercizio 3. L'algoritmo di insertion sort può essere espresso come una procedura ricorsiva in questo modo. Per ordinare un vettore A di n elementi ($A[1..n]$), prima ordiniamo il vettore corrispondente ai primi $n-1$ elementi ($A[1..n-1]$) e poi inseriamo l'elemento $A[n]$ nel vettore ordinato $A[1..n-1]$. Scrivere l'equazione ricorrente per il tempo di ordinamento e calcolarne la complessità.

Esercizio 4. Disegnare l'albero di ricerca binario (*BST*) che si ottiene inserendo le seguenti lettere,

A L G O R I T H M S

una ad una (lette da destra a sinistra), in un albero di ricerca binario vuoto. Mostrare quindi la sequenza di alberi binari che si ottengono cancellando una ad una le seguenti lettere

A L G O

Esercizio 5. Scrivere un algoritmo che legge da tastiera una sequenza di n numeri interi con valori da 0 a k (compreso) e in seguito sia in grado di calcolare in un tempo $O(1)$ quanti elementi sono compresi in un intervallo $[a,b]$ specificato dall'utente. La complessità dell'algoritmo deve essere $\Theta(n+k)$.

