

Esame informatica III 2 febbraio 2010

1. Record di attivazione

Scrivi una funzione in C che dato in ingresso un array di interi, restituisce il numero di interi dispari in esso. Scrivi tre versioni: una non ricorsiva, una ricorsiva e una ricorsiva con tail recursion.

Specifica esattamente i parametri che passi alla procedura, il tipo di passaggio utilizzato e il loro significato.

Scrivi anche un main di esempio in cui chiami le funzioni con un array di tua scelta con 3 elementi. L'array deve essere dichiarato come variabile globale. Nel main deve essere dichiarata la variabile locale **Ndis** a cui assegnare il risultato della funzione di cui sopra.

Disegna il record di attivazione per tutte e tre le versioni fino alla massima estensione del record di attivazione. Nel caso di tail recursion, spiega quali ottimizzazioni hai adottato o potresti adottare.

2. Riferimenti in C++

Considera il seguente codice. Cosa stampa a video?

```
#include <iostream>

int main ()
{
    int val2 = 0;

    int foo = 999;
    int &ref1 = foo;
    cout << "foo = " << foo << ", ref1 = " << ref1 << endl;

    ref1++;
    cout << "foo = " << foo << ", ref1 = " << ref1 << endl;

    int &ref2 = ref1;
    ref2++;
    cout << "foo = " << foo << ", ref1 = " << ref1
        << ", ref2 = " << ref2 << endl;

    ref1 = val2;

    ref1++;
    cout << "foo = " << foo << ", ref1 = " << ref1
        << ", ref2 = " << ref2 << ", val2 = " << val2 << endl;
}
```

3. Cyclone

Riscrivi uno dei metodi implementati nell'esercizio 1 usando Cyclone. Valuta diverse alternative nel passaggio dei parametri.

4. Costruttori C/C++

Data una classe Foo che ha come costruttori:

```
Foo();  
Foo(int x);
```

Le seguenti istruzioni sono corrette? Quale costruttore vien invocato?

```
Foo t1;  
Foo t1();  
Foo t2 = Foo(8);  
Foo t2(8);  
Foo t2 = 8;  
Foo t2 = (Foo) 8;
```

```
Foo t3 = Foo(8);
```

5. C++

Che tipi di ereditarietà permette il C++? Per cosa si differenziano. Fai un piccolo esempio.

6. Dinamic Binding in Java

Considera le seguenti classi

```
class Z{}  
class Scritto { void stampa(Object o){}}  
class Libro extends Scritto { void stampa(Libro l){}}  
class Monografia extends Libro { void stampa(Object l){}}
```

la istruzioni (numerate per comodità) – assumi che siano eseguite in sequenza, e che se qualcuna non è corretta, non venga eseguita:

1. Object z = new Z();
2. Object ol = new Libro();
3. Scritto sl = new Libro();
4. Libro ll = new Libro();
5. Libro ls = new Scritto();
6. Scritto s1 = (Scritto) ls;
7. Scritto s2 = (Scritto) ll;
8. Libro m = new Scritto();
9. Object om = new Monografia();
10. Scritto sm = new Monografia();
11. sl.stampa(z);
12. sl.stampa(ll);
13. sl.stampa(ol);
14. om.stampa(ol);
15. sm.stampa(ol);
16. sm.stampa(ll);

sono corrette (se no perchè?) Quale codice chiamo nelle diverse chiamate del metodo stampa (S, L,M) ? Descrivi in dettaglio i meccanismi di selezione del metodo corretto.

7. Java

Cosa vuol dire che i tipi degli array sono covarianti? Che problemi può dare? Anche le List generiche (cioè List<A>) della JCF sono covarianti? E le List non generiche (con Object) hanno problemi simili?

8. Semantica assiomatica

Considera il seguente programma che dovrebbe calcolare in m il numero maggiore tra x e y:

```
m = 0;
while (y > 0 or x >0) {
    y = y - 1;
    x = x - 1;
    m = m + 1;
}
```

Alcune SOLUZIONI

Come scriveresti le precondizioni e postcondizioni? Come dimostreresti la sua correttezza? Se vuoi puoi usare la funzione $\max(a,b)$ che vale il massimo tra a e b (ma non è strettamente necessario).

Foo t1; -> ok c1

Foo t1 (); -> ERRORE

Foo t2 = Foo(8); -> costruttore 2

Foo t2(8); -> c2

Foo t2 = 8; -> c2

Foo t2 = (Foo) 8; -> OK c2