Politecnico di Milano
Facoltà di Ingegneria dell'Informazione
Informatica 3
Proff. Ghezzi, Lanzi, Matera e Morzenti
Appello del 18 Luglio 2005
Recupero I Parte

COGNOME E NOME (IN STAMPATELLO)

MATRICOLA

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.

Sia T una classe e ST una sua sottoclasse. In un ipotetico linguaggio L le variabili siano solo di tipo automatico, allocate nello stack. Si supponga anche che L consenta *assegnamenti polimorfi*. Ad esempio:

```
T a; ST b;
. . .
a = b;
```

Spiegare quali vincoli L dovrà imporre sulle definizioni di sottoclasse affinché possano coesistere polimorfismo e allocazione sullo stack.

Risposta:
Il compilatore allochera` spazio per la variabile a in base al suo tipo statico (T). Gli oggetti di tipo dinamico ST che potranno essere assegnati ad a non potranno occupare piu` memoria di quella allocata dal compilatore. Pertanto una sottoclasse non deve poter aggiungere nuovi attributi (variabili). Potrà solo aggiungere nuovi metodi e/o ridefinirne alcuni.

Esercizio 2.

Dato il seguente codice Java:

```
Class Confronto
{
  public static void main (String[] args)
  {
   String strA;
   String strB;

   strA = new String("Una stringa");
   strB = new String("Una stringa");

   if ( strA == strB )
    System.out.println("Stringhe uguali.");
  }
}
```

a. Scrivere l'output generato, fornendo una breve spiegazione.
b. Evidenziare eventuali problemi legati all'uso dell'operatore "==" per il confronto di uguaglianza tra oggetti e illustrare brevemente possibili soluzioni.

Risposta:
Il programma non stampa alcun messaggio. Nonostante i due oggetti `strA` e `strB` abbiano valori equivalenti, il loro confronto restituisce valore `false`. Il confronto viene infatti effettuato tra i soli riferimenti, e `strA` e `strB` memorizzano riferimenti a oggetti diversi.

L'operatore "`= =`" esegue quindi il confronto tra riferimenti ad oggetti, e non tra "valori profondi".
Nel caso in cui sia necessario il confronto tra valori, è per esempio possibile utilizzare il metodo `equals()`, disponibile per la classe `Object`, opportunamente ridefinito per effettuare anche il confronto tra valori.

Esercizio 3.

Quesito 1.
Si consideri il seguente frammento (incompleto) di programma in un ipotetico linguaggio C-like, che però consente il passaggio di funzioni come parametri di funzioni:

```
typedef
  struct {
    float b[5];
    int a;
} T;

int z;

void h (function p) {
      int c;
      c = p(z); (***)
      . . .
}

void f (T x) {
      int c;
      int g (int k){
            c = . . .; (**)
            h(g); (****)
            . . .
      }
      c = z + x.a (*)
      . . .
      h(g);
}

main()
{
      T y;
      ...
      f(y);
      ...

}
```

Calcolare le coppie <distanza, offset> per c, z e x.a in (*) e c in (**). Si assuma che nei record di attivazione le variabili siano allocate a partire da offset 3 (a offset 0 sia memorizzato il link dinamico, a offset 1 il link statico e a offset 2 l'indirizzo dell'istruzione di ritorno).

Risposta:
Assumo che gli interi occupino 1 parola e i float occupino due parole di memoria. Assumo che le variabili locali corrispondenti ai parametri formali siano i primi elementi memorizzati nel recordi di attivazione di una funzione e assumo che il link statico all'area dati globale punti alla prima variabile globale.

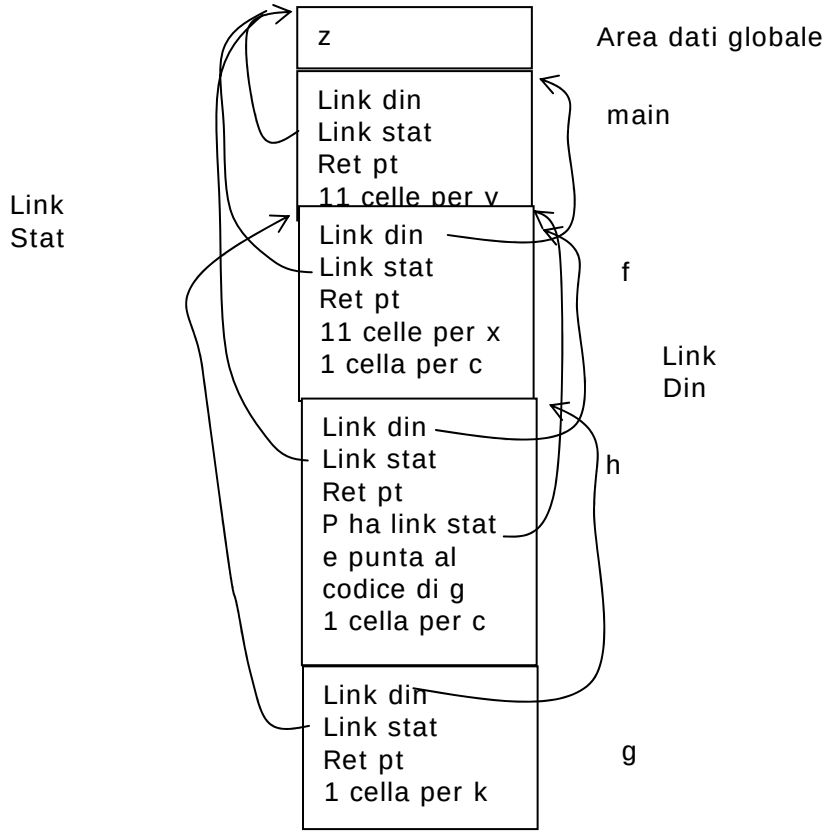
In (*): c <0,14> z <1, 0> x.a <0, 13>

In (**) c <1, 14>

Esercizio 3 (continua).

Quesito 2.
Considerare lo stato della macchina astratta immediatamente dopo l'esecuzione dell'istruzione (***) ipotizzando un'esecuzione del programma in cui h è chiamata da f. Illustrare dettagliatamente la sequenza dei record di attivazione e i relativi link statici e dinamici.

Risposta:



Esercizio 3 (continua).

Quesito 3.
Si continui l'esecuzione precedente con la chiamata h(g) in (****). Come nel precedente quesito, si illustri dettagliatamente la sequenza dei record di attivazione e i relativi link statici e dinamici dopo che h in (***) chiama p(z).

Risposta: