

Esercizi di Linguaggi di Programmazione

Luca Bedogni

March 12, 2015

1. Dato il seguente pseudocodice, dire cosa viene stampato sia nel caso si utilizzi un linguaggio con scope statico, sia nel caso in cui venga usato scope dinamico, quando si chiama A().

```
{
  int x = 2
  void A() {
    write(x)
  }
  void B() {
    int x = 10
    A()
  }
  B()
}
```

2. Dato il seguente pseudocodice, dire cosa viene stampato sia nel caso si utilizzi un linguaggio con scope statico, sia nel caso in cui venga usato scope dinamico, quando si chiama A().

```
{
  void A() {
    int x = 0
    void B() {
      int x = 10
    }
    {
      void C() {
        int x = 1
        B()
        write(x)
      }
    }
    C()
  }
}
```

3. Considerare il seguente pseudocodice

```

{
  int x = ?
  void A() {
    int y = ?
    write(x)
  }
  {
    void B() {
      int x = 1
      A()
    }
  }

  int y = 2
  B()
}

```

Fornire una prima configurazione di valori per x e y di modo che il programma stampi il valore 2, usando scope statico. Fornirne poi una seconda per stampare 2 utilizzando scope dinamico.

4. Dato il seguente pseudocodice, dire cosa viene stampato sia nel caso si utilizzi un linguaggio con scope statico, sia nel caso in cui venga usato scope dinamico.

```

{
  int x = 1
  void F(y, z) {
    int x = 0
    y = y + 1
    z = z + 1
    write(x+y+z)
  }

  F(x,x)
}

```

5. Considerare il seguente pseudocodice

```

{
  int x = 10
  int v = 5

  {
    void B() {
      write(x)
    }
  }
  {
    void A(z) {
      int x = z * v
    }
  }
}

```

```

        B()
    }
}

?
}

```

Fornire una chiamata alla funzione A di modo che il programma, usando scope statico, stampi il valore 10. Fornire una seconda chiamata per stampare il valore 10, ma usando scope dinamico.

6. Considerare il seguente pseudocodice

```

{
  int x = 1
  int y = 2

  void A() {
    int x = 2
    int k = 3
  }
  void B() {
    int x = 5
    A()
    ** 1 **
    x = 2
    C()
  }
  void C() {
    int z = 5
    ** 2 **
  }
  B()
}

```

Scrivere qual'è il contenuto delle variabili attive al punto **** 1 **** e **** 2 ****, utilizzando scope dinamico e scope statico.

7. Dato il seguente pseudocodice, dire cosa viene stampato sia nel caso si utilizzi un linguaggio con scope statico, sia nel caso in cui venga usato scope dinamico, quando si chiama A().

```

{
  int x = 2
  void A() {
    int x = 5
    void B() {
      int x = 6
      {
        int x = 5
        void C() {

```

```

        write(x)
    }
}
C()
}
}

```

8. Considerare il seguente pseudocodice

```

{
  int x = 12
  void B() {
    write(x)
  }

  {
    int x = 6
    void A(int z) {
      x = x * z
      B()
    }
  }

  ?
}

```

Fornire una chiamata alla funzione A di modo che il programma, usando scope statico, stampi il valore 12. Fornire una seconda chiamata per stampare il valore 12, ma usando scope dinamico.

9. Si vuole gestire lo heap con lista libera con blocchi a dimensioni variabili. Si progetti con un certo livello di dettaglio:

- Tutte le informazioni di gestione da mantenere all'interno di ogni blocco
- Qualora necessarie, le strutture dati ausiliarie per gestire con una certa efficienza una politica di allocazione first-fit e deallocazione che preveda la fusione di blocchi adiacenti.

Si consideri di avere 100000 posizioni di memoria nello heap, e di avere 5 programmi (P1, P2, P3, P4 e P5), che richiedono uno spazio così definito:

- P1: 3000 posizioni.
- P2: 5000 posizioni.
- P3: 40000 posizioni.
- P4: 20000 posizioni.
- P5: 7000 posizioni.

Si considerino i seguenti eventi:

1. P1 richiede di essere allocato
2. P2 richiede di essere allocato
3. P3 richiede di essere allocato
4. P1 termina
5. P4 richiede di essere allocato
6. P2 termina
7. P5 richiede di essere allocato
8. P3 termina
9. P4 termina
10. P5 termina

Schematizzare lo stato dello heap al termine dell'esecuzione dei punti 3,4,5,6,7,10, avendo cura di definire lo stato delle informazioni ausiliarie per ogni blocco.

10. Si consideri il seguente pseudocodice.

```
{
  int x = 5
  void foo ( int y ){
    x = x + 1
    y = y + 10
    x = x + y
    write (x)
  }
  {
    int x = 0
    foo (x)
    write (x)
  }
}
```

Dire cosa viene stampato sia nel caso si utilizzi scope statico sia nel caso in cui si utilizzi scope dinamico.

11. Si consideri il seguente pseudocodice.

```
{
  void a () {
    void b () {
      corpo_di_b
    }
    void c () {
      void d () {
        corpo_di_d
      }
    }
  }
}
```

```
        corpo_di_c
    }
    corpo_di_a
}
}
```

Fornire uno schema del display in seguito alla chiamata di procedure **a**, **b**, **c**, **d**, **b**, considerando che alla fine della successione di chiamate tutte le procedure siano ancora attive, ovvero nessuna di esse ha ancora ritornato il controllo.