



# LINGUAGGI DI PROGRAMMAZIONE

Esercitazione 14/12/2020



# Esercizio 1

Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome e scope statico (si ricordi che l'espressione `x++` restituisce il valore della variabile `x` e successivamente incrementa `x` di uno).

```
int x = 3;
void foo(name int y) {
    int x = 5;
    int w;
    x = x + y;
    w = y;
    write(x);
    write(y);
}
foo(x++);
write(x);
```

# Esercizio 1

Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome e scope statico (si ricordi che l'espressione `x++` restituisce il valore della variabile `x` e successivamente incrementa `x` di uno).

```
int x = 3;
void foo(name int y) {
    int x = 5;
    int w;
    x = x + y; x++
    w = y; x++
    write(x);
    write(y); x++
}
foo(x++);
write(x);
```

8  
5  
6

x = 3 | 4 | 5 | 6

|           |
|-----------|
| y = x++   |
| x = 5   8 |
| w = 4     |

# Esercizio 2

(i) Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope statico.

```
int x = 3;
int y = 4;
void foo(reference int y, reference int z) {
    int x = 5;
    y = y+1;
    if (z==y) write(x);
    else write (y);
}
foo(x,x);
write(x);
write(y);
```

(ii) Si dica poi cosa stampa lo stesso frammento se il passaggio dei parametri avviene per valore-risultato.

# Esercizio 2 (i)

(i) Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope statico.

```
int x = 3;
int y = 4;
void foo(reference int y, reference int z) {
    int x = 5;
    y = y+1;
    if (z==y) write(x);
    else write (y);
}
foo(x,x);
write(x);
write(y);
```

x = 3 | 4  
y = 4

y = x  
z = x  
x = 5

5  
4  
4

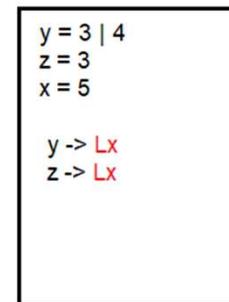
(ii) Si dica poi cosa stampa lo stesso frammento se il passaggio dei parametri avviene per valore-risultato.

## Esercizio 2 (ii)

(i) Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope statico.

```
int x = 3;
int y = 4;
void foo(reference int y, reference int z) {
    int x = 5;
    y = y+1;
    if (z==y) write(x);
    else write (y);
}
foo(x,x);
write(x);
write(y);
```

x = 3  
y = 4



4  
3  
4

(ii) Si dica poi cosa stampa lo stesso frammento se il passaggio dei parametri avviene per valore-risultato.

## Esercizio 2 (ii)

(i) Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope statico.

```
int x = 3;
int y = 4;
void foo(reference int y, reference int z) {
    int x = 5;
    y = y+1;
    if (z==y) write(x);
    else write (y);
}
foo(x,x);
write(x);
write(y);
```

x = 3 | 4  
y = 4

y = 3 | 4  
z = 3  
x = 5  
z -> Lx  
y -> Lx

4  
4  
4

(ii) Si dica poi cosa stampa lo stesso frammento se il passaggio dei parametri avviene per valore-risultato.

# Esercizio 3

Il seguente frammento di codice è scritto in uno pseudolinguaggio che ammette iterazione determinata espressa con il costrutto `for` che è implementato mediante iteration count. Si dica cosa viene stampato dal seguente frammento:

```
x=1;
for i=1 to 3+x by 1 do{
    write(i);
    x++;
}
write(x+1);
```

# Esercizio 3

Il seguente frammento di codice è scritto in uno pseudolinguaggio che ammette iterazione determinata espressa con il costrutto `for` che è implementato mediante iteration count. Si dica cosa viene stampato dal seguente frammento:

```
x=1;
for i=1 to 3+x by 1 do{
    write(i);
    x++;
}
write(x+1);
```

```
ic = 4
1  x = 2 ic = 3
2  x = 3 ic = 2
3  x = 4 ic = 1
4  x = 5 ic = 0
6
```

# Esercizio 4

Cosa stampa il seguente frammento di codice in un linguaggio con scope statico e passaggio per nome, e che valuta le sue espressioni da sinistra a destra?

```
int x = 3;
int y = 4;
int f(name int a){
    int x = 5;
    y = y+a;
    y = y+a;
    return a;
}
write (f(x+y) + y);
```

# Esercizio 4

Cosa stampa il seguente frammento di codice in un linguaggio con scope statico e passaggio per nome, e che valuta le sue espressioni da sinistra a destra?

```
int x = 3;
int y = 4;
int f(name int a){
    int x = 5;
    y = y+a; x+y      y = 11
    y = y+a; x+y      y = 25
    return a; x+y
}
write (f(x+y) + y);
```

53

# Esercizio 5

Si consideri il seguente frammento di codice:

```
int x = 5;
int y = 0
int pippo(name int m)
  {int x = 2;
   if m < 0 and x/y > 0 then return 100
   else return m + x
  }
write( x + P(x++));
```

Si dica se è possibile che l'esecuzione di tale frammento stampi il valore 13 e, se sì, sotto quali ipotesi.

# Esercizio 5

Si consideri il seguente frammento di codice:

```
int x = 5;
int y = 0
int pippo(name int m)
  {int x = 2;
   if m < 0 and x/y > 0 then return 100
   else return m + x
  }
write( x + P(x++));
```

- valutazione lazy
- espressioni valutate da sinistra a destra

Si dica se è possibile che l'esecuzione di tale frammento stampi il valore 13 e, se sì, sotto quali ipotesi.

# Esercizio 6

In un certo linguaggio gli assegnamenti sono valutati nel seguente modo: prima l'espressione a destra del simbolo =; poi quella a sinistra; infine si esegue l'assegnamento. Si consideri il seguente frammento di codice:

```
int x = 1;
int A[5];
for (int i=0; i<5; i++)
    A[i] = i;
A[x++] = A[x++]+1;
```

Qual è lo stato del vettore A dopo l'assegnamento ?

# Esercizio 7

Si consideri il seguente frammento di codice

```
int x = 5;

void pippo(int x, int y){
    int z = 0;
    z = x+1-y+y;
    write(z);
}
{pippo (*, *);
write (x);
}
```

Si definiscano le modalità di passaggio di parametri di pippo e si indichino i parametri attuali da sostituire agli \*, in modo tale che i valori stampati dal codice siano 7 e 8 (in quest'ordine).

# Esercizio 7

Si consideri il seguente frammento di codice

```
int x = 5;

void pippo(int x, int y){
    int z = 0;
    z = x+1-y+y;
    write(z);
}
{pippo (*, *);
write (x);
}
```

pippo (x++, x++)  
passaggio per nome

Si definiscano le modalità di passaggio di parametri di pippo e si indichino i parametri attuali da sostituire agli \*, in modo tale che i valori stampati dal codice siano 7 e 8 (in quest'ordine).

# Esercizio 8

Si consideri il seguente frammento di codice:

```
{int x,v;
  x = 10;
  v = 5;

int fie(int w,z){
    x = (w++) + z;
    write(x)
}

{x = 1;

*****

}
```

Si scriva al posto degli asterischi una chiamata a `fie` e si formulino delle opportune ipotesi in modo che venga stampato il valore 3. La chiamata deve usare come parametri attuali delle variabili.

# Esercizio 8

Si consideri il seguente frammento di codice:

```
{int x,v;  
x = 10;  
v = 5;
```

```
int fie(int w,z){  
    x = (w++) + z;  
    write(x)  
}
```

```
{x = 1;
```

```
*****
```

```
}
```

passaggio per nome  
valutazione da sinistra a destra

Si scriva al posto degli asterischi una chiamata a `fie` e si formulino delle opportune ipotesi in modo che venga stampato il valore 3. La chiamata deve usare come parametri attuali delle variabili.