

# Fondamenti di Informatica, A.A. 2013-2014

29/05/2014

## Esercizio 1

È dato il frammento di codice Matlab

```
n = 2;
x = ones(1,10);

x = x*2;

for i=1:n
    x = x.^2;
end

y = x > 6;

disp(y)
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

## Soluzione

Il vettore `x` prima dell'inizio del ciclo `for` contiene 10 elementi tutti pari a 2; nel ciclo `for` ciascun elemento viene elevato al quadrato per due volte, quindi il risultato viene confrontato con il valore 6. Pertanto l'interprete Matlab stampa

```
1    1    1    1    1    1    1    1    1    1
```

## Esercizio 2

Discutere quando è appropriato utilizzare un ciclo `for` e quando un ciclo `while`. Riportare almeno un esempio per entrambi i casi.

## Soluzione

Il ciclo `for` viene utilizzato quando si conosce a-priori il numero di iterazioni da eseguire. Un tipico esempio in cui l'utilizzo di ciclo `for` è appropriato è la scansione di un array. Il ciclo `while` è indicato quando si richiede il verificarsi di una condizione specifica la quale può bensì essere il raggiungimento di un numero ben preciso di iterazioni; tuttavia il suo utilizzo è indicato quando non si conosce a-priori il numero di iterazioni da eseguire ma si vuole proseguire fino al raggiungimento di una condizione di tipo differente (raggiungimento di una soglia di tolleranza d'errore, verificarsi di uno specifico evento, etc...).

## Esercizio 3

In una prova di esame di informatica viene richiesto di calcolare l'equivalente decimale di un numero binario rappresentato nell'array `a` in stile LittleEndian (vale a dire che le cifre

sono memorizzate nell'array dalla meno significativa alla più significativa). Uno studente propone il seguente codice

```
n = 0;

l = length(a);

for i=1:l
    n = n + (2.^ i).*a(i);
end
```

Si chiede di trovare l'errore presente nel codice e di proporre una correzione adeguata.

### **Soluzione**

Il problema consiste nell'assenza dell'esponente 0. Una possibile soluzione è la seguente:

```
n = 0;

l = length(a);

for i=1:l
    n = n + (2.^(i-1)).*a(i);
end
```

#### Esercizio 4

È data la seguente funzione Matlab

```
function [x n]=simple(a,x,b)
    n=length(x);
    for i=1:n
        x = a\b;
        n = length(x)-1;
        x = x.^2;
    end
end
```

Si stimi il numero di operazioni aritmetiche eseguite, assumendo che  $a$  sia una matrice triangolare inferiore  $n \times n$  e che  $x$  e  $b$  siano vettori  $n \times 1$ .

#### Soluzione

Il comando `a\b` risolve un sistema lineare; essendo  $a$  triangolare inferiore richiede  $n^2$  operazioni aritmetiche. Essendo eseguito  $n$  volte si avranno  $n^3$  operazioni aritmetiche. Verranno inoltre eseguite  $n$  sottrazioni ottenendo così un totale di  $n^3 + n$  operazioni aritmetiche. Infine, per ogni iterazione, si eseguono  $n$  elevamenti a potenza ottenendo  $n^3 + n + n^2$ . Da notare che la definizione della variabile  $n$  all'interno del ciclo non influenza il numero di iterazioni eseguite.

#### Esercizio 5

È dato il frammento di codice Matlab

```
n=100;
maxt = 30000;
v = [];
x = 1:10;
t = 0;

i = 1; j = 1;
while i<=n && t<=maxt
    while j<i
        v(i) = j+i;
        j = j+2;
        t = max(v);
    end
    j = 1;
    i = i+2;
end
```

Riscrivere il codice facendo uso di uno o più cicli `for`.

## Soluzione

Una possibile soluzione è la seguente:

```
n=100;
i = 1; j = 1;
maxt = 30000;
v = [];
x = 1:10;
t=0;

for i=1:2:n
    for j = 1:2:i-1
        v(i) = j+i
        t = max(v);
    end
    if(t > maxt)
        break;
    end
end
```